



Makers All

How to Help Communities
Transform Emerging Tech
So They Can Shape Their Destinies

Anders Schneiderman
May 2019

Table of Contents

Executive Summary	2
Introduction	7
Part 1 Smoothing the Learning Curve	
1.1 Create a Culture of Community-Oriented Coding UX	16
1.2 Create a Continuum Of Skill	21
1.3 Evangelize Breakthrough Research on Coding UX	25
1.4 Create Institutional Support For Coding UX	29
Part 2 Create a Community-Based Ecosystem of Support	
2.1 Ground Tech Education In The Community	32
2.2 Build A Better Bridge Between Training And Work	38
Part 3 Integrate Tech Training and Civic Engagement Training	
3.1 Why Training In Civic Engagement?	43
3.2 What Citizenship Schools Can Teach Us	45
3.3 Applying The Lessons Of Citizenship Schools	48
3.4 How Emerging Tech Will Turn The Economy Upside Down	52
Conclusion	56

EXECUTIVE SUMMARY

We are at a crossroads.

Robots and Artificial Intelligence (AI) threaten to either put millions out of work or create a society so unequal it shatters our democracy.

At the same time, we are facing a once-in-a-century opportunity. Over the next 20 years, not only robots and AI but virtual and augmented reality, digital fabrication, and other emerging tech will become ubiquitous. Emerging tech will create an abundance of wealth, and it could become a vital part of revitalizing communities from Compton to Appalachia.

But there's a catch. To take advantage of this opportunity, we need to pull off the seemingly impossible: train as many people as possible in every community to become developers and designers, so they can either get jobs or create co-ops or small businesses in emerging tech.

How do we do that? By recognizing that the road to a better future runs through our past.

What Extension Services and Citizenship Schools Can Teach Us

Extension Services

In the late 19th and early 20th century, U.S. farming faced a daunting task. The modern agricultural revolution couldn't occur unless millions of farmers mastered a wide range of new technical knowledge and skills. After several efforts failed to have a big enough impact, the United States created a community-oriented approach called Cooperative Extension Services aka Extension Services. Extension Services was a smashing success, radically transforming the way millions of U.S. farmers practiced farming.

In the introduction, the report argues that the reason Extension Services succeeded was because in its most effective incarnations, it:

- **Made Modern Farming Techniques Much Easier to Learn** by breaking down complex agricultural research and designing farming techniques so they could be easily understood by everyday farmers, and by creating feedback loops between researchers and community networks of farmers
- **Fostered A Rich Community-Based Web Of Support**, including clubs, local farmer demonstrations, and other forms of peer-to-peer learning
- **Operated On A Massive Scale**, providing every agricultural county with one or more extension agents, who organized large numbers of volunteers and identified and developed local leadership
- **Provided Real Accountability** by focusing on states and counties and demonstrating whether extension agents were transforming farming in their designated community -- unlike many well-meaning coding initiatives today, which may track how many individuals have taken a coding class but never measure whether their efforts are bringing about fundamental change in specific communities.

The report argues that the history of Extension Services also helps explain why so many people think the idea of truly democratizing coding is just wishful thinking. In many communities in the US, smart, dedicated, and passionate people who operate on a shoestring, pouring their heart and soul into making programming accessible in their community. But they are hopelessly outmatched by the size of the problem. If we are going to succeed, we need a solution that operates at the same scale as and employs strategies as ambitious as Extension Services did.

Citizenship Schools

Although the Extension Services strategy has proven to be extremely effective at helping the audiences it's designed to serve, sometimes it's done tremendous harm by focusing on the needs of some audiences while ignoring or damaging others -- e.g., helping white farmers while ignoring black farmers, turning its back on the small family farms it once embraced while shoring up Big Ag and damaging the environment.

How do we harness the strengths of the best traditions of Extension Services while ensuring no communities are left behind? By embracing the lessons of 1960s Civil Rights Movement's Citizenship Schools.

Citizenship Schools were a little-known but crucial part of the 1960s Civil Rights Movement. Southern states had passed voter suppression laws that required voters to be able to read and write. To teach basic literacy, Citizenship Schools used community-oriented strategies similar to those of Extension Services. But they also combined basic literacy with civic literacy, teaching their students the skills they needed to fight for their freedom and for their community.

When the Internet first took off, it was sold as a tool for empowering everyone; instead, it became one critical foundation of an economy where communities from Harlem to Harlan County were left behind. If we don't want to repeat that mistake, we must ensure that every community has a seat at the table -- and to do that, we'll need to draw on the lessons of Citizenship Schools. Only by ensuring that there are enough people in every community who understand both how to code and how to fight for their community's future can we be confident that emerging tech opportunities will be accessible in every community and that every community will have a real say in who benefits from this new economy.

3 Strategies for Truly Democratizing Emerging Tech

While the exact approach taken by each community will differ -- the Bronx, rural Kentucky, East LA and Youngstown, Ohio don't face identical challenges, circumstances, or culture -- the report argues there are three overarching strategies that are crucial to democratizing emerging tech in every community.

1) Smooth the Learning Curve

Why does training lots of people to become coders seem like Mission Impossible? Because right now, coding can be painfully hard to learn. But that's not a feature, it's a bug.

The techniques for making coding easier already exist -- user experience design (UX). UX is routinely used to make websites and apps easier to use and to entice users to buy more junk or spend more time clicking the Like button. But when it comes the coding languages and frameworks that are used to build those websites and apps? Not so much.

We need to do for emerging tech coding what Extension Services did for ag tech: make it a top priority to design it so it's easier for everyday adults to learn. Part 1 explains how:

- **Embrace Community-Oriented Coding UX** so we expand coding beyond the people who already find it easy to learn. Focus on the needs of everyday adults by creating collaborations between tech companies and community groups to improve coding UX. And in doing so, create coding UX expertise in the community that can lead to jobs or small businesses.
- **Create A Continuum Of Skill** from beginners to power users to "blue-collar coders" to skilled developers, so there are more opportunities to get into the industry. Then use coding UX to smooth the learning curve along that continuum. And along the way, strive to turn all types of coders from tool users to tool makers; as the report argues in "Hip-Hop Wasn't Created by Sound Engineers," you don't always need to be a rocket scientist to power innovation.
- **Get Coding UX Research Out Of Its Silo** so breakthrough techniques get out of the lab and into the hands of the coders who are creating and maintaining programming languages and frameworks.
- **Create Institutional Support For Coding UX**, particularly in Big Tech companies, venture capitalists, universities, and foundations.

2) Develop an Ecosystem of Community-Oriented Support

Making coding easier to learn is only half the battle; we also need to reduce the gap between tech and communities. In Part 2, the report argues that to do so we need to create an ecosystem of community-oriented support:

A) **Harness the Power of Community** in the way we provide training and support. For example:

- **Create Multiple On-Ramps**, especially in community institutions such as churches or beauty parlors, where people can explore the idea of coding and put their toes in the water while surrounded by people they know and trust
- **Create Peer-To-Peer Support** and other forms of community-based learning in order to overcome insecurity, psychological barriers -- e.g., seeing oneself as someone who works with their hands -- and uncertainty
- **Explore Collaborations between Communities and Corporations** in creating a rich ecosystem of support that can address overlapping needs, helping to both underwrite community support and create informal connections that can lead to opportunities for jobs and creating small businesses
- **Operate at Scale**, using national nonprofit networks and other approaches to take advantage of combining resources across communities

B) Build A Better Bridge Between Training And Work so all the hard work and sacrifice that goes into getting trained pays off. The report discusses strategies ranging from exploring creating apprenticeships to helping rural and other communities where tech jobs are scarce find ways to kickstart their local emerging tech economy.

3) Combine Tech and Civic Engagement Training

In Part 3, the report argues that emerging tech will not only create new economic opportunities, it will upend some of our core assumptions about how markets work. As it does so, it'll open up possibilities to reshape the rules of the road so our economy starts working for everyone. But if we want more than a handful of people to be involved in these conversations, we must ensure that as many people as possible in every community learn both the technical and the civic skills needed to truly participate.

This strategy must be designed so it can assist people from a wide variety of political ideologies and perspectives learn how to shape the direction of emerging tech, our communities, and our society. In short, the goal isn't to push a specific political viewpoint, it's to help revitalize our democracy in an era of rapid technological and economic change.

In Part 3, the report:

- **Explains how emerging tech will transform our economy over the next 20-30 years**, requiring that communities foster the skills of civic engagement around tech -- including addressing issues about the impact of emerging tech that go far beyond good paying coding jobs
- **Explores the lessons of Citizenship Schools**, which played a critical role in the 1960s Civil Rights movement in the South by combining basic literacy and civic literacy
- **Explores how to translate Citizenship Schools' lessons** for emerging tech by describing two thought experiments about how we could embed the development of civic literacy skills in technical training and support

Conclusion

In the conclusion, the report explains how to:

- **Get Started** implementing these ideas, including how to get your feet wet, how to set goals so you can know if you're making real progress, and how to ensure your efforts help to create diversity both in your community's emerging tech scene and the emerging tech industry as a whole
- **Scale Up** to create institutions and a movement that operates at a similar size and scope as Extension Services
- **Build Bridges** between communities in red and blue states

With the rise of emerging tech, we have a rare opportunity to help locked out communities today and create an amazing future tomorrow. But to seize it, the report argues, we need to learn the most important lesson from Extension Services and Citizenship Schools: we need to stop being afraid. We need to think boldly, dream, and then act on a big enough scale to reach those dreams.

Introduction

The Threat

We live in an age of a technological paradox. The computing power that used to fill a floor now fits inside your pocket, and the Internet and other new technology have produced staggering amounts of wealth. But despite claims that this technological revolution would empower everyone, wages for most workers have stagnated for decades, and communities from Harlem to Harlan County have essentially been written off.

The future could be even more grim. Many experts believe that in the next 10-20 years, 25-75% of all jobs will be eliminated by robots and Artificial Intelligence (AI). If they're right, this crisis could end up devastating the middle class and the poor and destroying our democracy.

And if they're wrong? We still aren't off the hook. A recent McKinsey report argues that by 2030 every advanced economy will need to "retrain and redeploy tens of millions of midcareer, middle-age workers."

How big is that challenge? In terms of magnitude, it's akin to coping with the large-scale shift from agricultural work to manufacturing that occurred in the early 20th century in North America and Europe, and more recently in China.

But in terms of who must find new jobs, we are moving into uncharted territory. Those **earlier workforce transformations took place over many decades**, allowing older workers to retire and new entrants to the workforce to transition to the growing industries. But the speed of change today is potentially faster. The task confronting every economy, particularly advanced economies, will likely be to retrain and redeploy tens of millions of midcareer, middle-age workers. As the [McKinsey Global Institute] report notes, "**there are few precedents in which societies have successfully retrained such large numbers of people**" (emphasis added).¹

A challenge of this magnitude would be scary and exhausting if our society only had to tackle it once. But McKinsey's prediction just takes us to 2030. As robots and AI continue to improve, we can expect wave after wave of mass dislocation as both old and new jobs are eliminated by new rounds of automation.

Policies that support "lifelong learning," active labor market policies, and programs such as Universal Basic Income could soften this series of blows. But ultimately they're like buying better lifeboats for the Titanic.

Regardless of whether robots and AI lead to mass unemployment or mass displacement, we are facing a daunting future.

The Opportunity: The Explosion of Emerging Tech

Our future doesn't have to be grim. Even as robots and AI's impact on work may pose a serious threat to our future, there is another technological trend that will give us a once-in-a-century opportunity to make our communities whole.

Over the next 20 years, not only robots and AI but also augmented and virtual reality, digital fabrication, and other emerging technologies will become ubiquitous, becoming one of the core driving forces in our economy. As they do so, they will create an abundance of wealth.

Robots and AI threaten to shatter the link between that wealth and broad prosperity: new industries may not create lots of good jobs. But if we can ensure everyday people get a seat at the table, if we can figure out how to train millions of people from Compton to Appalachia to become developers and designers, they should be able to capture a big enough slice of emerging tech's wealth to help revitalize our communities.

As we'll see in Part 3 and the conclusion, this opportunity won't solve all of the economic problems created by robots and AI -- e.g., not everyone is going to become a programmer or designer. But if we take advantage of this opportunity, it could serve as one of the central foundations for rebuilding our communities.

Is Truly Democratizing Emerging Tech a Pipedream?

But can we train millions of everyday people to become developers and designers? Given where we are today, it's understandable that many think this goal is just wishful thinking.

It's not for lack of trying:

- Every day around the globe, people who are smart, dedicated, and passionate -- and who usually operate on a shoestring -- pour their heart and soul into making computer tech accessible in their community
- Many people in the tech industry work hard to create tools and languages they hope will empower millions of people
- Educators in academia and the tech industry have spent countless hours developing freely available online coding classes in order to reduce the barriers for everyday people to learn programming

This impressive work has made a real difference, often opening up possibilities for new careers and new journeys of self-discovery for the individuals they've helped.

But go to communities facing hard times and ask people if they think coding will create a lot of opportunities that people in their community can count on. You won't find many takers.

The problem we face is not that the people trying to democratize coding aren't driven enough or passionate enough or smart enough. The problem is that they are hopelessly outmatched by the scale and scope of the challenge. If emerging tech is to provide real hope for communities that our society has left behind, we need a new approach.

Luckily, we don't have to figure out a solution from scratch. This isn't the first time our society has faced the need to bring about a massive transformation of the technical skills and knowledge of millions of people. The road to our future runs through our past.

Lessons from Our Past: Cooperative Extension Services and Citizenship Schools

Cooperative Extension Services

In the late 19th and early 20th century, the US faced a similarly daunting task. A modern society can't function unless farmers are productive enough so they can feed the vast majority of people who no longer work the land. To pull off this agricultural revolution, millions of farmers had to master new skills and knowledge -- the key elements of soil science, plant science, entomology, and a wide range of other information and practices that made up modern farming practices.

From a distance, teaching farmers about modern farming practices might seem much easier than teaching programming to people who've never written a line of code. But the gap between what most farmers did and modern agricultural was so large and difficult to bridge that it took more than 50 years of large-scale experiments to figure out how to do it.

For example, after many years of volunteer-driven efforts to spread modern practices, in 1862 Congress and the Senate reluctantly concluded that a more dramatic government intervention was needed. The Morrill Act created a system of land-grant colleges to teach agriculture and other practical arts. The Act was driven in part by the belief that the practice of modern agriculture was so complex that only by helping many people gain a college degree in it could the US transform the world of farming. But these and other efforts failed to train enough farmers to substantially change farming practices.

Finally, in 1914, the US came up with a strategy that worked. Drawing on the lessons learned from years of experimentation by individual states, the Smith Lever Act created a community-oriented approach called Cooperative Extension Services.²

Cooperative Extension Services were successful because they created a rich web of support to help farmers make the transformation to modern farming. In doing so, they employed four strategies:

- **Focus on Communities, Not Just Individuals.** Most of today's efforts at democratizing tech are focused on individuals. And although this approach has some advantages, it often ends up masking the fact that these efforts are creating opportunities in some communities but not others. Extension Services used a community-oriented approach, which asked whether communities, not just individuals, were succeeding in adopting modern agricultural practices.
- **Harness the Power of Community.** As we'll see in Part 2, by being deeply embedded in communities, Extension Services was able to leverage a community's assets, including the bonds of friendship and support among farmers, to have a far greater impact than they

otherwise could have. Without drawing on each community's strengths, it's doubtful that Extension Services could have succeeded.

- **Move the Tech Closer to the People.** Extension Services agents not only helped translate complex scientific concepts so they were accessible to ordinary farmers, they also helped create a feedback loop that ensured that the tools and practices proposed by researchers were modified so they fit farmers' needs and incorporated their experiences.
- **Operate at Scale.** Extension Services operated in every agricultural community across the US. Every county had one or more extension agents who fostered this rich web of support, most of whom were backed up by faculty and staff from their state's Land Grant colleges. And in a significant number of states, the scope of Extension Services' operation was remarkable. In New York State, for example, by 1948 Extension Services had built a network of 32,000 trained volunteer local leaders and committee members, who were supported by 383 agricultural and home economics staff affiliated with state colleges and universities.³

The Limitations of Extension Services

But Cooperative Extension Services also has a more complicated lesson to teach us. Extension Services has demonstrated that it's a remarkably effective way to serve a specific audience. However, at different times and in different places in the US (and around the globe), it has often been designed to help some audiences while inflicting terrible harm on others. For example:

- Throughout much of its history it actively discriminated against African-American farmers and ignored the needs of immigrant farm labor
- Over time, as it began to embrace the ethos of "get big or get out," it increasingly focused on the needs of Big Ag to the detriment of the small farmers it was originally designed to serve

At the same time, the most bottom-up, grassroots-oriented traditions of Extension Services have become so important to today's efforts to reduce global poverty that in 2015, Bill and Melinda Gates argued that

investing in extension so that it helps more farmers in more places—women as well as men, smallholders as well as more commercial farmers—is the only way to reap the full benefit of innovation.⁴

How do we embrace the best parts of Extension Services' traditions and avoid the worst? By learning the lessons of another remarkable educational effort: the 1960s Civil Rights Movement's Citizenship Schools.

Citizenship Schools

One of the major challenges facing the 1960s Civil Rights Movement was how to overcome voter suppression laws designed to stop African-Americans from voting by requiring that voters must be literate. How could the movement help African-Americans throughout the rural, agriculturally-dominated South to learn to read and write in a relatively short period of time so they could build political power? The solution: Citizenship Schools.

As we will see in Part 3, Citizenship Schools played a vital role in the success of the Civil Rights Movement. Like Extension Services, Citizenship Schools were designed so they were deeply rooted in their communities -- a critical step if they were going to help people overcome their

feelings of shame about being illiterate and their fears of violent retaliation. And like Extension Services, they operated at scale: a total of 1,000 Citizenship Schools were set up throughout the Deep South.

Where Citizenship Schools differed from Extension Services is that Citizenship Schools were deeply rooted in civic literacy and activism. While some traditions of Extension Services were grounded in civic engagement, Citizenship Schools were designed from the ground up to help their students learn how to fight for their freedom and for their community.⁵ In addition to teaching basic literacy, Citizenship Schools taught their students the civic skills necessary to win the struggle for voting rights, as well as to understand the nuts and bolts of how the political system worked so students could use political campaigns and community organizing to make their voices count.

When the Internet first took off, it was sold as a tool for empowering everyone; instead, it became one critical foundation of an economy where communities from Harlem to Harlan County were left behind. If we don't want to repeat that mistake, we must ensure that every community has a seat at the table -- and to do that, we'll need to draw on the lessons of Citizenship Schools. Only by ensuring that there are enough people in every community who understand both how to code and how to fight for their community's future can we be confident that emerging tech opportunities will be accessible in every community and that every community will have a real say in who benefits from this new economy.

Letting Go of Fear

Ultimately, the most important lesson we can learn from Extension Services and Citizenship Schools is that we can't be afraid to think big.

We have a rare opportunity to make more emerging tech jobs more accessible today and to create a better future for all tomorrow. But to seize it, we need to think boldly, dream, and then act on a large enough scale to realize those dreams.

Applying the Lessons of the Past to Build A Better Future

From West Baltimore to Letcher County, Kentucky to East LA to Youngstown, Ohio, every community faces distinct challenges and circumstances that will require a unique set of solutions. But there are some approaches that will provide the foundation to any community's approach. This report argues that the following three strategies are the key to our overall success:

1) Smooth the Learning Curve

For most emerging tech, there is a chasm of knowledge between its users and its creators, and the learning curve between the two is intimidatingly steep. Extension Services faced a similar challenge in the gap between the technical knowledge of agricultural researchers and farmers. The most effective Extension Service programs figured out how to bridge the gap by creating a feedback loop between farming communities and academia that made cutting-edge farming techniques accessible and practical. Part 1 argues we'll need to similarly revolutionize emerging

tech so it's easier for everyday people to learn emerging tech basics and to move along the path to mastery.

To do that, we need to take the "user experience" (UX) design movement that's transformed modern web design and adapt it to the world of coding. This includes:

- Create a tech culture of community-oriented coding UX design
- Create a continuum of skill from beginners to power users to skilled developers and use UX design to smooth the learning curve along that continuum
- Get academic research on coding UX out of its silo so new breakthrough techniques for radically reducing the time it takes to learn coding can become widely used
- Create institutional support for coding UX, particularly with big tech companies, venture capital (VC), foundations, and universities

2) Develop an Ecosystem of Community-Oriented Support

Making coding easier to learn is only half the battle; we also need to reduce the gap between tech and communities. In Part 2, we will show how to do so by:

- Changing the way we provide training and ongoing support so it harnesses the power of community
- Building a better bridge between training and work, from exploring apprenticeships, to helping rural communities where tech jobs are scarce, to find ways to kickstart their local emerging tech economy
- Doing all of this in a way that will both honor the unique issues and concerns of each community while operating on a scale comparable to Extension Services.

3) Creating a Community Tech Culture of Civic Engagement

Emerging tech will not only create new opportunities for jobs and creating co-ops and small businesses, it will open up possibilities for fundamentally rethinking how the economy works -- and with it a chance to reshape its rules so the economy starts working for everyone. To take advantage of these possibilities, we will need many people in every community who are fluent in emerging tech, not only because some of these struggles will revolve around highly technical issues, but also because some of the most important decisions will get made not in the halls of Congress or state legislatures but through formal and informal industry standards and best practices.

This strategy should not promote a particular ideology but instead aid people from different political backgrounds, communities, and identities/perspectives to help shape the direction of emerging tech, our communities, and our society. In short, the goal isn't a specific policy outcome, it's to assist in revitalizing our democracy in an era of rapid technological and economic change.

In Part 3, the report:

- Explains why the way emerging tech will affect our economy over the next 20-30 years will require that communities foster the skills of civic engagement around tech
- Explores methods for embedding the development of civic literacy skills in technical training and support

In the conclusion, the report will outline how people in the tech world and people who are active in their community might use or adapt these ideas to build a better future.

Implementing these strategies will require a lot of hard work, but **it's not rocket science**. Similar strategies have already been tested out in the real world by Extension Services and Citizenship Schools. All we need to do is to figure out how to translate them so they can address emerging tech's needs.

Report Scope

Adults

This report is focused on addressing the needs of adults, not children. There are three reasons for this decision:

- 1) From Black Girls Code to code.org to AI4ALL, there are already many groups doing terrific work focused on the needs of children.
- 2) When it comes to children, it's less necessary to advocate for the report's ideas because some of these ideas are already being implemented. For example:
 - As we will see in Part 1, there's plenty of work being done on improving coding UX for kids
 - Most children are already enmeshed in a variety of community networks, from school to after-school clubs to church programs, and some of these networks are encouraging kids to get involved in coding and design
- 3) Adults and children have very different needs. For example:
 - If you're introducing a nine-year-old to coding, you can count it as a success if they get a coding job 13 years in the future. Adults need coding jobs today.
 - Adult learners face very different emotional challenges than children do. If you've been out of work or have had a series of low-paying jobs or have lost a good paying job to automation, learning something as complex as coding can seem overwhelming.
 - Adults also face different financial challenges than most children do -- e.g., while they are learning they also have to put food on the table.

The Robot/AI Threat to Jobs

This report is agnostic as to whether robots and AI will lead to mass unemployment. While McKinsey studies are better than a Magic 8 Ball, there's simply no way to know who's right. Instead, this report is based on the assumption that we should stop obsessing over trying to predict the future and start focusing on creating a strategy to build a more just, prosperous economy regardless of the impact robots/AI have on jobs.

Furthermore, this report is **not** an attempt to build a comprehensive version of that strategy. As mentioned earlier in this chapter, even if our society becomes far more effective at training many people in every community to become developers, it's unrealistic to expect that everyone will do so. As the report discusses in Part 3 and the Conclusion, we will need other approaches

to ensure that everyone can lead a life with dignity and security and that all communities can be made whole. This report is simply an effort to lay out an approach for building one important piece -- but only one piece -- of the broader strategy our society needs.

The US

This report is focused on the circumstances facing the US. But the ideas and techniques it proposes could certainly be modified to fit the needs and circumstances of other countries.

Version 0.1

The problems we face in attempting to truly democratize emergent tech are far too complex for any one person or small group to have all the answers. The goal of this report is not to provide a definitive solution but to start a conversation about what it'll take to democratize emerging tech on a much larger scale.

Part 1

Smooth the Learning Curve

1.1 CREATE A CULTURE OF COMMUNITY-ORIENTED CODING UX

Consider the path that brought you to this report. Perhaps you found a link to it on Twitter or Facebook. Or you discovered it through a search you did on Google. Or you found it through a WordPress.com blog. However you got to it, there's a good chance that the technologies that routed you to the words in front of you were shaped by experts in a methodology that helps them figure out what makes a target audience tick and then design a website or app so it's easy for that audience to use. That methodology is user experience design, also known as UX.

In the worlds of web and mobile, UX has had a deep and profound impact on how designers work. You can take courses in UX design, go to conferences about it, even get a job doing it. If a corporation, nonprofit, or government is serious about reaching their audience, they've got staff or consultants who live and breathe UX. As you know from surfing the web, not every site uses UX design. But for anybody serious about building a great front-end, UX is a critical tool in their arsenal.

But for the programming languages and frameworks developers use to build these front-ends, it's a different story. The people who design these languages and frameworks rarely apply UX techniques to make it easier for your average person to be a developer. You don't need to spend much time reading code that looks like this:

```
if (hits.length) {  
  const hit = hits[0];  
  const hitMatrix = new THREE.Matrix4().fromArray(hit.hitMatrix);  
  this.model.position.setFromMatrixPosition(hitMatrix);  
  DemoUtils.lookAtOnY(this.model, this.camera);  
  this.scene.add(this.model);  
}6
```

to know that UX focused on everyday adults did not grace this house.

There is one major exception: programming languages and environments designed for young kids. There's a mountain of research on how to make it easier for young children to learn to code. Some of the resulting languages/tools, such as Scratch, are quite impressive.

But for the languages and frameworks adults need to know to get a job, the UX revolution might as well not exist.

Today, most efforts at making emerging tech accessible focus on training people to use coding tech as it is. In short, they try to move people closer to the tech. Despite a lot of hard work by very dedicated people, this strategy isn't enough to close the gap. Learning to code is still too hard and too clunky to have any chance of becoming a major opportunity for many people in every community. To truly democratize emerging tech, we also need to work the other side of the equation: move the tech closer to the people.

What Is UX and How Do You Do It?

What is UX?

Although there are many ways to define UX, one of the simplest and best definitions comes from the Nielsen Norman Group, whose founders were UX pioneers:

The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use.⁷

A core feature of UX is the concept of usability:

Usability is a quality attribute that assesses how easy user interfaces are to use....

Usability is defined by 5 quality components:

- **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency:** Once users have learned the design, how quickly can they perform tasks?
- **Memorability:** When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- **Errors:** How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction:** How pleasant is it to use the design?⁸

Doing UX Doesn't Have To Be Complicated

Bringing UX to coding might sound like a lot of work, but it doesn't have to be. For example, here's how the Nielsen Norman Group describes how to use one of the most popular techniques, user testing, to fit a user interface to a particular audience's needs:

- Get hold of some representative users...
- Ask the users to perform representative tasks with the design.
- Observe what the users do, where they succeed, and where they have difficulties with the user interface.

...

To identify a design's most important usability problems, **testing 5 users is typically enough. Rather than run a big, expensive study**, it's a better use of resources to **run many small tests** and revise the design between each one so you can fix the usability flaws as you identify them. Iterative design is the best way to increase the quality of user experience. The more versions and interface ideas you test with users, the better (emphasis added).

UX can be a complex, labor-intensive practice. But even a small amount smartly done can make a big difference.

Why Coding Tool Makers Don't Realize Their Tech Is Inaccessible

Today, the UX of coding libraries, frameworks, etc. is usually tested by seeing if the tool makes sense to a tech company's coders and other people with a similar background. So it's no surprise that these coders don't realize their tools are unnecessarily intimidating or inaccessible to people who aren't like them.

For example, one of the major development platforms for creating virtual reality and augmented reality is the Unity gaming engine. The company that built Unity has developed a large number of tutorials to help people master Unity development. In their first tutorial for beginners, here's the first script you write:

```
public class PlayerController : MonoBehaviour {
    public float speed;
    private Rigidbody rb;

    void Start () {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate () {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");
        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);
        rb.AddForce (movement * speed);
    }
}
```

The people at Unity who developed this script, who are undoubtedly passionate about democratizing Unity's tech, didn't recognize that scripts like this might intimidate many beginners and might pose a significant obstacle.¹

This is completely understandable. Unity has fostered a wonderful, thriving user community, that includes plenty of beginners who've picked up the basics of the Unity framework. If you're part of this community, scripts like the one above might not seem like a big deal.

Similarly, the developers of many programming libraries and frameworks are skilled at listening to their existing users and then evolving their tools so they get better and better at serving these user's needs. What they've accomplished using this strategy is often quite impressive.

But these approaches have a serious flaw: self-selection bias. Despite good intentions, these developers aren't addressing the needs of people who aren't using their tool -- especially people who find their tool too intimidating. If the tech world wants to truly democratize emerging tech, we're going to need to think beyond the user experience of the people we are already reaching.

¹ Wouldn't it be better to skip scripting altogether and use a visually-oriented approach? The jury's still out. As fast.ai's cofounder Jeremy Howard argues, visual approaches are often "more awkward, takes longer, and you see less on the screen at once... The hard bit of data science is not the typing, it's knowing what to write."
<https://www.theverge.com/2018/6/12/17452742/deep-learning-ai-learn-lobe-made-easy-coding>

Tech company CEOs say they want to empower not just people like themselves but everyone. That means empowering truck drivers and waitresses, hairdressers and janitors, ex-miners and nursing home aides. If they're serious about doing that, they need to "level up" and focus on these audiences.

Community-Oriented UX: Coding UX's "Special Sauce"

If a tech company randomly picked five people off the street who weren't geeks and did a little UX testing of their coding platform, they'd be way ahead of the game. At this point, any efforts to move beyond their usual audience would yield useful results.

But if tech companies want to truly democratize emerging tech, they should try a more ambitious strategy: building relationships with community groups, especially those who are already providing technical training.

One of the reasons why millions of US farmers were able to master modern farming is because Extension Services created a feedback loop between researchers and farming communities. If an agricultural researcher had devised a new approach for reducing problems with a particularly pesky beetle or a better strategy for planting wheat, an Extension Services agent would show it to farmers in their county. They'd get feedback from the farmers about what worked and what didn't, which would get passed back to researchers.

There's no reason that the tech industry couldn't begin to create similar relationships with community groups around coding UX. These community groups could provide a great environment for not only conducting formal UX studies but also providing ongoing real-world feedback that benefited both communities and the tech industry.

In doing so, it could offer communities additional benefits:

- **Small Businesses Community Co-Ops For Coding UX Services.** Community groups could create small businesses or co-ops that provide the tech industry one-stop shopping for UX testing services. In doing so, they would also create relationships that might lead to future business opportunities.
- **Early Access to Cutting-Edge Hardware.** As part of the UX studies, tech companies would give some members of marginalized communities early access to emerging tech hardware that they otherwise couldn't afford. And that in turn could give people in these communities a chance to get involved in emerging tech in an early stage where mastering the tech before it's widespread can open up new opportunities.
- **From Tool Users to Tool Makers.** Community-oriented coding UX could shift the perspective of coding beginners. Rather than seeing themselves as people who just use tools, they would also be involved in making them. Community groups could even explore modifying their trainings so that part of the learning process is learning how to evaluate and envision improving the coding experience for the next generation of coders in their community.

Help Communities, Help Your Bottom Line

Tech company CEOs are passionate about democratizing their technology because they care about society. But if they do it right, this approach will also help their company's bottom line.

Emerging tech's accessibility isn't just an issue for marginalized communities. Corporations and large nonprofits also struggle with how hard it is to develop solutions using emerging tech. Corporations can often paper over the problem by throwing money at it, spending scary amounts of money on consultants and/or hiring techies to do work that regular staff ought to be able to handle if the tech was properly designed. But if it's easier for people in South Central or Harlan County to master emerging tech, it'll also be easier for corporate staff to master it.

And that could be a major competitive advantage for any companies competing in a new tech market. In the battle to see which companies will win the biggest slices of an emerging tech pie, smoothing the learning curve could dramatically speed up the adoption of their tools.

1.2 CREATE A CONTINUUM OF SKILL

So far, we've focused on making the skills of a full-blown developer as accessible as possible. But is that level of skill really necessary for every job?

This report argues that if we want to increase the odds of truly democratizing emerging tech, in every field of emerging tech we need to ask two questions:

- Is it possible to create a continuum of skill in this field -- e.g., from beginners to power users to blue-collar coders to highly skilled developers?
- Can we use coding UX to reduce the work required to "level up" along this continuum?

From Power Users to Blue-Collar Coders

Power Users

In most organizations, IT staff and their consultants spend a lot of time building systems that let users produce powerful analyses with a click of a button. And yet it's not uncommon to find plenty of users in these organizations who do most of their analysis with Microsoft Excel.

One-click solutions are great so long as users stick to the well-marked trails the solution supports. But what if your needs take you off that trail? For example, what if you need to make changes to the report's format that the solution doesn't support? You're hopelessly stuck without the help of a developer.

But with Excel, if you're a little adventurous it's remarkable what you can do as a "power user." You can start by learning a few tricks, then gradually add more tools to your toolbox as you need them. The results aren't always pretty; sometimes it feels like Excel is duct tape for data. But like duct tape, you don't need to be highly skilled to use it to solve a wide range of problems.

If we're going to create an economy where lots of people in communities from Harlem to Harlan County can make a living from emerging tech, we need to start designing emerging tech tools so they can do what Excel does so well: support a culture of power users. Power users don't need a computer science college degree -- or for that matter any college degree -- to possess valuable skills that pay well. In short, by designing tools that support power users, the tech industry can create a wider range of opportunities for paid tech work.

Blue-Collar Coders

On the other end of the spectrum, Anil Dash and others argue that not all programming jobs should require a full-blown background in computer science. Instead, we need to foster the development of "blue-collar coders."

Vocational-technical schools (vo-tech) provide trained workers in important fields such as healthcare, construction trades, and core business functions like accounting. For a significant number of my high school peers, vo-tech was the best path to a professional job that would pay well over the duration of an

entire career. Now it's time that vo-tech programs broadly add internet and web technologies to the mix. We need web dev vo-tech...

Put another way, our industry can grow in a very meaningful way by giving lots of young people at a high school level the knowledge they need to learn [web development frameworks] straight out of high school, or teaching maintenance on a MySQL database at a trade school without having to get a graduate degree in computer science.⁹

Exactly what blue-collar coding will look like will differ from field to field. But the more we can create jobs that fall along a continuum of skill, the more opportunities we have to open up emerging tech development to communities our society has left behind.

Smooth the Learning Curve along the Continuum

Recently, several tech companies have been exploring the space along the continuum of skills. They've created "low code" or "no code" tools aimed at what some have called "citizen developers."

These tools are often quite powerful. But most of them suffer from the same problem: if your needs outstrip what the tool was designed to do, the learning curve to level up your skills is too steep.

If we aren't careful, there's a real danger that these well-meaning efforts to empower people will lead to the creation of a new class of dead-end jobs. That's why any effort to create a continuum of skill needs to prioritize smoothing the learning curve along that continuum.

Create a Continuum for Artists and Designers As Well As Coders

Emerging tech won't just require coders. The tech industry will also need designers and artists for AR/VR, robots, 3D printing/digital fabrication, and other forms of emerging tech. It's worth exploring if we can also create a continuum of skill for this work -- especially since, as is already the case with web design, some of this work may require the ability to do a little coding.

Opportunities to make a living from designer and artist work could play a particularly important role in marginalized communities. In the past, these communities have often been founts of new forms of art and culture. Both they and our society overall will greatly benefit if they are able to fully participate in shaping the world of emerging tech.

With Automation, a Continuum of Skill Is Even More Critical

When people talk about the threat of robots/AI, they often sound like they assume automation is a one-time event. It won't be. We can expect wave after wave of automation that either eliminates or deskills jobs.

And we can't assume that the pace of automation will slow down. If anything, it's at least as likely that over time the pace of automation will speed up.

In the past, there was often a significant gap between rounds of mechanization because the tech needed to mechanize one skill didn't easily translate to another. Wheat harvesting, for example, was mechanized in the 1930s and 40s, but it took almost two decades before the mechanization of tomato harvesting became commonplace.

In contrast, it took just a few years for the machine learning libraries used to categorize pictures of cats on the Internet to begin to be used in a wide range of fields, from drones to medical systems that can analyze some types of x-rays better than most doctors. Thanks to the Internet and Open Source, we can expect even more cross-pollination in the future.

As AI/robotics allow people to do more and more with less skill, we can also expect it will lead to the rise of new, highly demanding developer skills. That's why creating a software tool development process that iteratively creates and re-creates a continuum of work is so critical.

To do so, we will need to continually ask two questions:

- As automation allows us to do more and more with less skill, how do we create more opportunities for power users?
- As developers continue to build ever more powerful new languages, frameworks, APIs, etc., how can they be designed so gaining more skill and more sophistication in using them is easier to do?

In short, as automation continues to transform the skill landscape, we will need continual rounds of user experience design to smooth the learning curve so "leveling up" is as easy and engaging as possible.

Hip-Hop Wasn't Created By Turntable Engineers

Some advocates for power users or blue-collar coders sound as if as if they think these types of workers won't be capable of the kind of creative work that will drive emerging tech forward. While it's certainly true that some cutting-edge emerging tech work will require a PhD, the history of hip-hop suggests you don't need to be a rocket scientist to power innovation.

Hip Hop came out of neighborhoods that had lost hundreds of thousands of jobs to outsourcing and had been devastated by urban renewal. And yet the people who lived in these neighborhoods built some of the most amazing artistic creations the world has ever seen, transforming both music and culture around the globe.

A key part of Hip Hop's rise was a brilliant technical innovation: morphing turntables from tools for playing music into tools for making music. This innovation didn't come from the engineers who created turntables, it came from people like Grandmaster Flash. Grandmaster Flash, one of the "holy Trinity" who created hip-hop, had the most technical training of the three, and he'd only attended a vocational high school. But while Grandmaster Flash wasn't a PhD scientist, he

was a mad scientist -- a genius with an obsessive drive to experiment and create something new.

As important as it is to spread the most advanced technical skills to every community, what the birth of hip-hop shows us is that we must also empower people with a wide range of technical skills if we want to unlock the full creative and economic potential of emerging tech.

1.3 EVANGELIZE BREAKTHROUGH RESEARCH ON CODING UX

The following are some examples of what researchers in University of Washington's Code & Cognition Lab have cooked up to make coding more accessible to all:¹⁰

- **Learning by Solving Debugging Puzzles.** Mike Lee created Gidget, an interactive tutor for learning the programming language Python by solving debugging puzzles. In a study of 1,000 adult learners, he found that:
 - People who used Gidget picked up Python twice as fast as a well-regarded online Python tutorial and learned twice as much Python as participants who learned by doing.
 - Gidget "changed attitudes about the difficulty of learning to code from negative to positive in 20 minutes" -- an incredibly important finding for efforts to make coding accessible.
- **Learning by Tracing on Paper.** Benji Xie believed learners often have a "brittle" knowledge of a programming language's semantics, so they just guess what will happen when their program runs -- a surefire strategy for producing frustration and failure. He developed an innovative strategy that involves students figuring out on paper how their program will execute. His study showed that with less than 15 minutes of training in this technique, students did 15% better on computer class lab problems and 7% better on midterms -- and no one who used the strategy failed the midterm, in contrast to 25% of the control group.
- **Speed Up Learning JavaScript.** Greg Nelson created PLTutor, an online interactive textbook designed to cover all of the basics of standard JavaScript *in 3 hours*. For each core idea in JavaScript, PLTutor explains why and when you'd want to use that idea, then lets you step through code using the idea so you can see how the code executes and what impact it has. His study demonstrated that compared to a well-regarded online class, participants who used PLTutor had 60% higher learning gains.

The Code and Cognition Lab has even more dazzling work in the pipeline. In a 2018 talk, Professor Andrew Ko says they are now working on a project to create a scalable online tutor that "provides infinite personalized practice by applying program synthesis and our theories of programming knowledge." Their goal: students would be able to master the basics of a 10 week intro computer science course in just 10 hours.

There are many more research shops around the globe, many of which have produced equally impressive work.

And if you're a developer or a coding trainer, you've probably never heard of any of them.

In fact, it's possible to spend an entire career as a coding professional, reading articles and blog posts in your area of specialization, going to tech conferences, and talking with your colleagues, without ever bumping the world of research on making coding easier to learn.

It's not that some researchers aren't working with people out in the field. For example, Gidget influenced some of code.org's work. But the broader connections between academia and

practitioners rarely exist. Too often this research is like a tree that falls in a forest and nobody but researchers hear it. If we're going to truly democratize emerging tech, that's got to change.

The following are some strategies researchers might try if they want their research to make some noise.

WARNING: this chapter gets into the (geeky) weeds, so if you aren't a techie and don't work in tech, you may want to skip to Chapter 2.4.

1) Embrace Tech Groups

More researchers need to start working closely with groups within the tech world. For example:

- **Pythonistas.** The Python programming language is often used for teaching beginners because it is one of the easier languages to learn, and there are many people in the community of "Pythonistas" who are passionate about spreading Python. If researchers could build a bridge to these people, they might find a very receptive audience for the results of the research. They also might find people who would be interested in helping them with their research.
- **JavaScript Evangelists.** The world of JavaScript has undergone several transformations in the past decade or so. If researchers were an active part of that subcommunity, they might find strategic moments where they could help shape its direction. And there plenty of websites for web designers and developers, such as Smashing Magazine and CodePen, where researchers could find influential voices in the JavaScript community who are receptive to research on making coding easier -- after all, they're already strong advocates for website UX.
- **Tech Organizations for Women and People of Color.** The past decade has also seen the rise of a wide range of tech groups aimed at making coding and other tech more accessible to women and people of color. Researchers who shared the same passions could undoubtedly find people receptive to their work, some of whom would also be interested in collaborating.

2) Focus More Research on Libraries/Frameworks

One of the most striking features of academic coding UX research is that it is primarily focused on languages -- either on modifying existing languages or on creating new ones. But these days, libraries, APIs, and frameworks that are built on top of programming languages are at least as important for accessibility than the underlying language.² For example:

- If someone says they are using Python for machine learning, odds are they aren't spending much time directly working with plain-vanilla Python. The bulk of their work is using Python

² Although there are important differences between libraries, APIs, and frameworks, for the purposes of this report you can treat them as interchangeable. The distinction that matters here is between a programming language, such as JavaScript or Python, and tools such as libraries, etc. that are built on top of a programming language in order to extend its capabilities.

libraries with names like Tensorflow or Pytorch, and they may also be using a Python library called pandas to "clean up" their data so it's ready for analysis.

- If someone is developing VR/AR, they probably aren't spending much time with plain-vanilla C# or JavaScript, they're mostly using a C# framework called Unity or a JavaScript library called A-Frame.

If researchers focused more of their time on influencing libraries and frameworks, they might have a bigger impact.

A) It's Easier to Change Frameworks/Libraries

- Getting Python coders to agree to a fundamental change to their language or to switch to a new language that's designed to be easier to learn is a pretty tall order. Changes in a language, for example, can end up breaking a staggering number of scripts and software products. Convincing them to switch from one Python data visualization library or machine learning library to the library researchers have helped make much more accessible? That's a much easier sell.
- It's also common for developers to switch libraries/framework as better ones become available, so there are more opportunities for researchers to get their ideas out. And if their UX ideas help propel the use of an up-and-coming library, other library/framework designers are more likely to adopt those ideas.

B) It's Easier to Build an Easy-to-Use Library on Top of another Library

- If users are already using an emerging tech library, it's often pretty straightforward to build another library on top. For example, fast.ai's library that underlies their terrific course on deep learning is built on top of the Python library Pytorch.¹¹ This strategy can be a great way to either eliminate the hard parts of the underlying library or shield users from its weirder bits when they are first getting started.
- This strategy lends itself to an iterative, organic approach, trying out a researcher's ideas and getting quick feedback -- or even working with members of the community to critique libraries as they are learning them and to provide suggestions for improving them.
- Researchers who build an easier to use library on top of an existing library also have an almost built in audience for their work, which improves the odds they'll have an impact.

3) Make a Communications Plan

When medical researchers publish their findings in a journal, either their team or their institution has at least an implicit communications plan. Sometimes it involves a sophisticated press strategy, including having PR staff attempting to get the researchers booked on TV and radio. At the very least they put out a press release.

But in the field of UX coding research, this appears to happen only rarely. Not surprisingly, the results of most UX coding research never reaches the audiences who need to know about it if it is going to have a real impact.

UX coding researchers don't need communications strategies as sophisticated as medical researchers; they aren't trying to reach a general audience. But even a little communications work could go a long way -- especially if researchers build relationships with tech groups.

4) Learn from Extension Services

Most UX coding researchers work in universities and colleges. And in many of those universities and colleges, there are staff who have a solid track record of making information more accessible: Extension Services. Coding and agriculture face different audiences and challenges. But there are surely valuable lessons UX coding researchers could learn from the decades of experience of their agricultural counterparts.

The Tech World Must Play Its Part

If researchers' work is going to have a significant impact, ultimately it's up to them to make it happen. But those of us who aren't researchers can also play a part by seeking out their research. Large tech companies can play a particularly important role, as they have the resources to find interesting research and figure out how to implement it.

And if university and college researchers begin to build a more vibrant relationship with the world outside academia, demonstrating the practical worth of their research to a wider audience, large tech companies could help fund research that might have a strategic impact.

For example, it's relatively difficult to get research funding on making machine learning easier to learn.¹² Given that most big tech companies are pouring huge amounts of money into machine learning/AI, it's in their self-interest to either fund this research themselves or lobby to get it funded.

1.4 CREATE INSTITUTIONAL SUPPORT FOR CODING UX

Say you work at a small startup or a big tech company and you're frantically racing to build the first version of a new emerging tech library. You want to democratize your tech and you care about the community. But you can barely keep your head above water. How would you find the bandwidth to take on community-oriented coding UX or creating a continuum of skill? Where would you even start?

Or perhaps, as many popular tech libraries first began, your project is something you work on after hours as a labor of love. If you eventually succeed, you'll probably have a small army of Open Source volunteers helping you out. But for now, you don't have the resources to take on community-oriented coding UX.

If we are going to truly democratize emerging tech, we need to bring about a cultural change in the tech industry so techniques for smoothing the learning curve are as common for emerging tech development as UX is for designing websites. Getting there will require institutional support. The following are some ideas about what that support might entail.

Large Tech Companies

For Google, Microsoft, and other large tech companies, there are two keys to making community-oriented coding UX and other techniques for smoothing the learning curve an inherent part of the process:

- **Resources.** For any emerging tech project that is creating languages, libraries, frameworks, etc., the company should ensure they have the resources they need to fully embrace community-oriented coding UX and create a continuum of skills -- e.g., UX staff who work on the project and/or a liaison with a community group that's providing the UX work.
- **Rewards.** Companies should ensure staff are rewarded for making community-oriented coding UX a priority.

Venture Capitalists

While we certainly wouldn't expect all venture capitalist funds to embrace techniques for smoothing the learning curve, if a significant number of them began expecting that their emerging tech startups would use this approach, they could radically transform the tech community.

Foundations

- **Fund UX Services for Small Fry.** Foundations could provide grants to community groups to provide free coding UX services. These services could be targeted at unpaid labors of love, very early startups, and other individuals or organizations that have a significant audience for their open source coding tool but don't have institutional support. Foundations might also facilitate the creation of a pool of these resources that's funded by foundations, big tech companies, and other actors.

- **Shape Software Project Grants.** In some cases, foundations directly fund software projects - e.g., Jupyter/IPython notebook, which has become a major development environment for data science and machine learning. They could begin to require that grantees follow a community-oriented coding UX approach and, if relevant, a strategy for fostering the creation of a continuum of skills.
- **Leverage Nonprofit Software Purchases.** Many nonprofits purchase software using foundation grants. If foundations banded together, they could have a significant impact on the ecosystem of tools used by their grantees.

Colleges and Universities

- **Shape Student Startup Culture.** For colleges and universities whose computer science departments teach their students how to create startups, if they encouraged community-oriented coding UX and other strategies for smoothing the learning curve, it could substantially speed up the adoption of these techniques.
- **Leverage University Resources.** Colleges and universities are especially well situated to have an impact, because not only do they train many of the students who will create startups and go to work at tech companies, but they also create research on UX techniques and coding UX. Many colleges have agriculture extension services as well as researchers and centers who have extensive experience collaborating with the community. If they leverage all of these resources, they can provide a much richer experience for their students, who will go on to help shape tech industry culture.

Part 2

Create a Community-Based Ecosystem of Support

2.1 GROUND TECH EDUCATION IN THE COMMUNITY

Heart disease kills more men in the United States than any other illness. High blood pressure is one of the biggest risk factors for it, and yet according to Doctor Aaron Carroll, "we have had a hard time getting patients to comply with recommendations and medications." A recent study targeting hard-to-reach patients tried to help them using a community-based strategy:

Barbers screened patients, then handed them off to pharmacists who met with customers in the barbershops. They treated patients with medications and lifestyle changes according to set protocols, then updated physicians on what they had done.

When one part of the community ran into trouble, another stepped up. For example, if some barbers were having trouble consistently measuring customers' blood pressure, pharmacists helped out.

The end result: in six months, 63% of the intervention group now had normal blood pressure compared to only 12% of the control group. The reason for this success?

Getting barbers involved meant health messages came from trusted members of the community. Locating the intervention in barbershops meant patients could receive care without inconvenience, with peer support. Using pharmacists meant that care could be delivered more efficiently.¹³

Similar experiments in other fields have been equally effective. And as we have seen with Extension Services and Citizenship Schools, when a community-oriented approach is used on a large scale, they can produce remarkable results.

In Part 1, we discussed how smoothing the learning curve can narrow the gap between emerging tech and people in the community. To close the remaining gap, we will need to harness the power of a community-oriented approach to education.

Today, most community-based groups for teaching coding use the often-meager resources they have to employ a few facets of a community-oriented approach to education. What if we provided the resources and institutional support they need so they could collectively operate on the same scale that Extension Services and Citizenship Schools did? The following is a sketch of what this approach might look like.

Pieces of the Community-Oriented Education Puzzle

Different types of communities will need a different set of strategies to help members of their community master emerging tech. But there are some strategies that can be incorporated by all communities. The following are some examples.

Create Multiple On-Ramps

For someone living in a community our society has written off, the idea of getting started in coding can be daunting. The obstacles they face can feel overwhelming. And if classes will put a

big strain on your life because of the cost and time they require, how do you know it's worth the sacrifice? At this point, you don't even know if you like coding enough to want to do it for a living.

That's why we need to create multiple on ramps. We need to go where people are, where they are surrounded by their peers, and create opportunities for them to get their feet wet.

Where it makes sense to create on ramps will depend on the community. For example:

- Barbershops and beauty salons
- Union halls
- Churches and other houses of worship, who could offer a short program before or after one of their weekly sermons so members could try a little coding and talk to people who are already doing it

Wherever people congregate, wherever they are surrounded by people they know and trust, we should explore ways of breaking down barriers and firing up interest.

Community-Oriented Trainings and Support

In the play *Our Town*, when George tells Emily about his plans to become a farmer, he says:

Y'know, Emily, whenever I meet a farmer I ask him if he thinks it's important to go to Agricultural School to be a good farmer. Yeah, and some of them say it's even a waste of time. You can get all of those things, anyway, out of the pamphlets the government sends out.

George is exaggerating a bit, but the idea wouldn't seem out of line to farmers in his time. Extension Services was created because the system of Land Grant colleges that had been built to help spread modern agricultural practices through formal schooling couldn't reach enough people. Extension Services succeeded where Land Grant colleges failed because in virtually every county, it embedded one or more extension agents who built upon existing social networks and created new ones to create a massive infrastructure of informal peer-to-peer learning.

Extension agents used a wide range of techniques to create an infrastructure for training and support:

- **Identified and Developed Natural Leaders.** Extension agents often focused on identifying and developing natural leaders: farmers who were already widely respected in their community. These natural leaders had preexisting social networks and relationships they could use to recruit other farmers. And they were also likely to understand the concerns and fears that agents needed to address if farmers were to be convinced to adopt new techniques.
- **Nurtured Neighborhood Clubs.** Extension agents helped farming communities form neighborhood clubs and worked with clubs to ensure farmers got a steady stream of ideas about how they could improve their farming. As a result, farmers weren't just hearing about an idea brought in by an outsider, they were learning while surrounded by their peers who spoke the same language and understood the realities they faced.
- **Produced Informal Learning Materials.** To supply these clubs, extension agents provided lots and lots of pamphlets and other written materials that were crafted to help teach

farmers new techniques and address any concerns they had. Many extension agents also were heavy users of radio broadcasts and other new forms of communication.

- **Fostered Community Events.** Extension agents helped foster state fairs and other places where farmers could see demonstrations, compete to see who could use new techniques to grow the best crops, etc.
- **Reached Adults through Their Children.** By creating and supporting 4-H Clubs, Extension agents not only began training the next generation of farmers, they also gave adult farmers the chance to learn from their children's successes with new techniques that have been proven to work not just in the laboratory but in the very fields they farmed.
- **Fill the Gaps.** As the story at the beginning of Part 2 demonstrated, any system of support is inevitably going to have gaps; being able to dynamically address these problems can make the difference between success and failure. Extension agents were expected to look for gaps and find solutions to fill them.

If we were to adapt the techniques used by Extension agents to the circumstances communities are facing today, we could substantially increase our ability to make emerging tech coding more accessible.

This kind of community-based approach also has the potential to knock down obstacles that many are afraid we can't overcome. For example, some have argued that many people -- especially men -- who work in blue collar jobs won't want to become coders because their sense of identity is tied to working with their hands. That's a huge problem for an outsider who's trying to convince them. But if someone who's resistant is being recruited by a natural leader they trust, and if they know they'll be making this transition surrounded by their peers who've also spent a lifetime working with their hands, this is a much easier nut to crack.

Building on Existing Efforts

To implement Extension Services' community-oriented approach, we don't have to start from scratch. From online courses to blogs and podcasts to terrific online communities like Glitch and CodePen to face-to-face Meetups and hackathons, we already have some of the elements we'll need for building a great informal, peer-oriented training and support system. Now we need to learn from the example of Extension Services about how to take this fledgling infrastructure to the next level.

For example, tech meetups are often a great educational resource for those who attend them. But even these meetups don't reach anywhere near the number of people needed to make a big difference. This is often because there simply aren't enough resources available to do the labor intensive work for community-oriented strategies to reach more people.

But sometimes this is also because tech meetups need to adapt their culture to these more ambitious goals. For example, for everyday people who live in a community where they don't know anyone who makes a living from emerging tech, these meetups often feel intimidating and alienating.

Similarly, some parts of the tech learning universe have cultures that are hostile to outsiders. For example, many beginning and experienced developers rely heavily on a Q&A website called

Stack Overflow as a major source for learning new tricks and troubleshooting problems. But the culture on Stack Overflow has gotten so bad that in 2018, Jay Hanlon, Stack Overflow's executive vice president of culture and experience, felt compelled to write a post called "Stack Overflow Isn't Very Welcoming. It's Time for That to Change." He confessed,

Too many people experience Stack Overflow as a hostile or elitist place, especially newer coders, women, people of color, and others in marginalized groups.

This is exactly the kind of problem that a community-oriented approach is designed to solve.

Creating Real Accountability

Although the immediate issue with Stack Overflow was its toxic culture, there was a deeper problem driving this issue. Stack Overflow wasn't unaware of the issues with its culture. As Hanlon explained,

Our employees and community have cared about this for a long time, but we've struggled to talk about it publicly or to sufficiently prioritize it in recent years. And results matter more than intentions.¹⁴

This points to a crucial difference between Extension Services and what the tech community has built so far: accountability. Extension Service agents were expected to deliver. Making progress in their county was their top priority. In emerging tech, with its patchwork of informal training and support, there's no one who's similarly accountable.

It's not that there isn't any accountability in emerging tech training and support. Ask the people who run an online training course how they're doing, and they're happy to give you a bevy of stats. But what they're measuring -- eg., how many students took their course -- doesn't tell us what we need to know. In an era where the tech world is mad for metrics, there is no one who can tell you if the patchwork of overlapping efforts of training and support have actually paid off for specific communities, helping enough people to get a job or create a business to make a real difference in that community.

It's too soon to tell if we need the equivalent of an emerging tech extension agent in every county. But what is clear from Extension Service's experience is that a key to success is accountability where it counts.

Designing In Diversity

As we struggle with the issue of accountability, we can also learn an important lesson from Extension Services' failures.

Extension Services often ignored African Americans or treated them like second class citizens. It often reinforced gender roles in a way that limited women's opportunities. Over time, it increasingly favored Big Ag over small family farms. The problem wasn't that Extension Services wasn't effective, it's that at points in its history it was designed to help some audiences and ignore or harm others.

This is why it's critical that as we build a community-oriented system of training and support, we must ensure the system we develop is designed from the ground up to be inclusive -- and it must hold people accountable for achieving this goal.

The Advantages of Operating at Scale

Operating at scale allows us to deploy resources and pool experience in a way that otherwise wouldn't be possible. For example:

Leveraging Our Collective Experience. Suppose a network of churches decided to work together to come up with the best strategies for helping members get their feet wet. One or two churches can begin an experiment, and other churches can learn from their experience, making it easier for smaller churches or churches whose resources are committed elsewhere to get involved. A network of churches can also make it easier for members with computer skills and tech volunteers to have a bigger impact than if they were working with just one church.

Leveraging Existing Resources

- If you are trying to build custom trainings it's much easier to fund it -- or get outside funders to pay for it -- if you can pool resources and talent from a large network.
- Suppose you need a bunch of computers to set up a training room or to give to low income students so they can work at home. It's common for midsize and large companies and nonprofits to replace their staff's computers every three years, which means they regularly have a large number of computers they no longer need. If you had a network of churches working together, it's much easier to get outside volunteers to help set up the hardware and to offer some ongoing support -- or, if it's a large network of churches, to get the funds to pay for it.

Explore Connecting Community and Workplace Support

Between waves of automation and waves of new tech, corporations are going to face a never-ending need to train and retrain their staff. Currently, most simply aren't equipped to do so.

Staff in corporations and other large organizations don't have the same needs as people in the community. But there are many areas where they may overlap. As communities are developing a rich ecosystem for their members, it may be worth exploring if there are ways to jointly address their needs. For example:

More Accessible Trainings for All. Trainings in a new technology are often intimidating for all but the most technically skilled staff. Because the tech is so new, the people who know it best are better at understanding the tech than they are at understanding how to teach beginners. But since community-oriented groups have to develop their emerging tech trainings so they're accessible for everyday people, there's a good chance corporate staff who aren't skilled techies would also benefit from these trainings. If corporations and other large local institutions invested in funding or providing staff to help develop these trainings, both the community and the world of work would benefit. There may even be ways to design trainings to facilitate this

approach -- e.g., building trainings that are like playlists, so it's easy to mix and match parts to fit a particular audience.

More Friendly and Inviting Support Groups. Many corporate and other organizational staff don't take advantage of tech meetups and other existing support groups because they find them intimidating. If a community is trying to create a more inviting support ecosystem, large organizations might find that their staff might also benefit from some facets of it.

By exploring these possible overlaps between corporations and communities, we can also get more bang for the buck:

- **Foster Informal Connections That Can Lead to Jobs, Opportunities to Create Wealth.** The ecosystem support could spawn informal connections between people in the community and people working in corporations and nonprofits. These kinds of informal personal connections are a great way of finding good jobs -- especially given that many jobs are never advertised. Similarly, these informal personal connections will be an invaluable resource for fledgling community co-ops and small businesses.
- **Leverage More Resources.** For corporations and other large organizations, embracing a community-oriented ecosystem support is a twofer. They get to give back to the community, and their organization directly benefits from the resource they have helped to create.
- **Provide a Bridge for Thinking through Learning-Work Connection.** The other function this connection could serve is to provide a space that will encourage thinking through the connection between learning, support, and work. That's the subject of the next chapter.

2.2 BUILD A BETTER BRIDGE BETWEEN TRAINING AND WORK

It takes a leap of faith as well as a lot of time and money to get the training you need to start a new career -- expenses that most folks in marginalized communities can barely afford. If there isn't a job or an opportunity to start a small business on the other end, it can dash hopes and drain pocketbooks, leaving people worse off than when they started.

This was the bitter experience many community groups had with green jobs. They worked hard to train members of their community only to discover that the jobs they'd been told to expect never materialized.

If we are going to ensure that as many people in every community can find work in emerging tech, training isn't enough. We need to build a bridge between training and work that communities can count on.

Connecting Training and Work When Jobs Are Plentiful

For inner-city and some other communities, there are often plenty of good paying emerging tech jobs in the surrounding area. The issue is how to ensure people in their community can get hired. The following are some strategies that might be worth exploring.

Apprenticeships

While training is valuable, what employers most want is experience. There's a big difference between what goes on in classes and the real world:

- **The Real World Is Messy.** In class, everything is designed to help you learn. If they're teaching you sophisticated AI techniques, for example, odds are you'll be working with cleaned up data. Classes usually spend relatively little time on what will eat up most of your day when you get a job: cleaning your data so it fits the AI technique's data requirements.
- **Tech Is about People.** In most classes, you spend most of your time working by yourself. In the real world, most developers work in teams, writing and maintaining code that's collectively owned. And many developers' jobs require them to work closely with users, juggle competing pressures from different departments, and occasionally deal with internal politics. These interpersonal skills, which can make the difference between success and failure, aren't taught in most tech classes.

And for prospective students, training is often expensive. You have to pay for the training, and you may only be able to work either part time or not at all.

This isn't a new problem. One solution that's worked well in several industries: apprenticeships.

Apprenticeships are rare in the tech world, but it may be time to start experimenting with them. Apprenticeships address both employers and trainees needs:

- Trainees get paid as they learn
- Employers end up with an employee with real-world experience

Make Job Description Work Requirements More Realistic

Today, many IT managers -- or their companies -- create job descriptions that don't fit their actual needs.

- **Credentials Creep.** In the field of machine learning, for example, it's not uncommon to see job listings that require a PhD even though the work the person would do in that job would almost never require that level of skill.
- **Expertise Inflation.** It's not uncommon for employers to ask for more years of experience than is needed -- occasionally for more years of experience than the tech has existed.

None of this is surprising. Managers and companies often don't have the knowledge or experience to know what level of expertise their organization needs, and they don't have an easy way to find out.

Inflated job requirements are a problem for all employees, but it's especially a problem for people who've taken a nontraditional path to obtain the skills employers need. They don't come from the kind of background most employers are most comfortable with. On top of that, whether they are white working class from the country or people of color from the inner city, they may face employers who consciously or unconsciously underestimate their abilities.

In short, if we want as many people in every community to have a real shot at emerging tech jobs, we need to start exploring how to ensure employers get the help they need so they don't ask for more expertise or experience than their jobs require.³

Foster Informal Connections

Many jobs are never advertised; people find out about them through informal networks. Although there are some structural solutions that might be worth considering to reduce the importance of these informal networks, we should also explore strategies for using networks to our advantage.

For example, as we discussed in the previous chapter, if we build a rich support ecosystem that spans community and work, we could use it to foster informal connections that could make employers more comfortable hiring people with nontraditional backgrounds and/or training.

Key Features of a Good Solution

As communities begin to experiment with solutions, here are a few things to keep in mind:

- **Don't Reinvent the Wheel.** We should always ask, what can we learn from other efforts to solve these problems? For example, what can we learn from union hiring halls and apprenticeships in other industries?
- **Address the User Experience of Trainees *and* Employers.** Efforts like these often don't grapple with what it'll be like for trainees to go through this process. And they almost always ignore the user experience of frontline managers. Even managers who want to do good in the world are often so overwhelmed with the rest of their job that idea of taking what they

³Some argue certificates are the solution. There are two problems with certificates: 1) for most employers, real-world experience is a far more useful gauge of a job candidate's skills, and 2) you can't create meaningful certifications for many emerging technologies because their standards and best practices are still in flux.

perceive to be a significant risk may be hard to do. Making it as easy and painless for both employers and would-be employees to participate can make the difference between success and failure.

- **Start with Low Risk, Expand the Circle.** Start experiments with employers who can afford to take some risks -- e.g., it's a lot easier to experiment with hiring if you're frequently hiring developers. But plan to invite in a wider range of employers as the project gains traction.
- **Focus on Diversity.** Although the tech world talks a good game about democratizing coding for all, most tech companies have made remarkably little progress in creating a workforce that is diverse as the society where they work. Therefore, any solution should be designed so it makes diversity a top priority.
- **Engage the Whole Public.** To have the greatest impact, we should tackle these problems as part of broader efforts at civic engagement (see Part 3). There may already be an initiative in your region to tackle these questions that most people in the community -- including many employers -- don't know anything about because the project didn't decide to deeply engage with the public.

What to Do If There Aren't Enough Jobs/Business Opportunities

The above strategies assume there are enough jobs or opportunities for creating small businesses in the region surrounding the community. But what if there aren't? What about, for example, rural communities where there may be few if any good paying tech jobs? The following are some strategies that might be worth exploring.

Use Large Tech Companies and Multinationals to Bootstrap the Process

Large tech companies and some large US multinationals have an almost insatiable need for tech staff. They couldn't supply enough jobs for every community that needs them. But they could certainly create pilot projects that might open up new opportunities.

For example, a few companies could begin by hiring small numbers of individuals and/or hire a fledgling consulting company from these communities. If these efforts were successful, they could experiment with scaling them up.

Over time this approach could help to bootstrap local tech scenes. For example, if a few large tech companies pave the way, showing that rural and other overlooked communities have the technical chops and the business/interpersonal skills needed to do a great job, other firms might be more willing to take a chance on hiring them.

There's no way to know in advance if this strategy would succeed on a scale that would be effective. And this approach would have to be careful not to substitute one problem for another -- e.g., making these communities overly dependent on large outside companies, whose needs could easily change. But given the needs of both parties, it's well worth trying.

Use the Government to Bootstrap the Process

If large tech companies and multinationals can't play this role in enough communities, we should explore having the government step in.

- When Silicon Valley first took off, it greatly benefited from strategically targeted government intervention. DARPA, other defense agencies, and intelligence agencies were often "early adopter" customers who were eager to buy new tech products before almost anyone else. In doing so, these federal government agencies created demand that helped kickstart new tech markets. There's no reason it couldn't use a similar approach aimed at marginalized communities.
- Like large corporations, the government has an enormous need for developers.

To figure out an effective role for the government, we would need to run some pilot projects. And there are number of potential dangers we would need to address -- e.g., to reduce the danger of corruption, political favoritism, nepotism, etc., we would need an approach that was fully transparent. But given the impressive track record the federal government has in helping underwrite the rise of the tech industry, if the private sector can't give rural communities a leg up, there's no reason why the public sector can't help.

Part 3

Integrate Tech Training and Civic Engagement Training

3.1 WHY TRAINING IN CIVIC ENGAGEMENT?

Neil Gershenfeld, director of MIT's Center for Bits and Atoms, oversees a wildly ambitious 50 year research project in digital fabrication whose ultimate goal is to create the equivalent of Star Trek's replicator. A few years ago, he had a surprising revelation: they'd figured out the technological roadmap to build replicators, and although it would take several decades to work their way down that road, the technical issues weren't the biggest hurdle.

We're finding we have to build a completely new kind of social order, and that social entrepreneurship—figuring out how you live, learn, work, play—is hard.¹⁵

As result, in 2017 he and his brothers Alan and Joel decided to write a book to kickstart a public debate about the impact of digital fabrication on society. They wanted to make sure we didn't make the same mistake we'd made with the Internet and personal computing.

As early as 1965, the signs of the coming digital revolutions [the Internet and personal computers/smartphones] were there for anyone to see. And yet most of the world missed them. As a result, few were prepared for the deep economic, social, and cultural impacts of the first two digital revolutions....

The negative aspects of the first two digital revolutions are not simply accidents. Nor were they driven by some unseen hand. Decisions made (and not made) and priorities set (and not set) early on, as the technologies were being developed and introduced to the market, have had lasting effects.

And if more than a handful of people are going to be involved in making those decisions about the digital fab revolution, we need to start a public debate now.

The best time to shape the destiny of transformative, accelerating technologies is early, before changes have become both widespread and entrenched. This is when the embedded assumptions in the technology and the initial market instantiations are in the early stages of formation and still negotiable.¹⁶

Digital fabrication isn't the only field that's going to create unprecedented change. In the next 20 years, robots and AI, augmented reality, and other emerging tech will begin to fundamentally transform our society and economy.

Imagine a future in which robots and AI, augmented and virtual reality, digital fabrication, wearables, and other emerging technologies have become ubiquitous. Because we've implemented the strategies described in Part 1 and Part 2, many people in every community are now working in emerging tech, developing designs for digitally fabricating sneakers or creating robot "recipes" for painting a home's interior. In this new economy, the answers to questions about how emerging tech markets should be structured could have a deep and profound impact on which communities and individuals prosper.

Today, only a handful of people have a say in answering questions like these. Equally importantly, few people understand the issues well enough to meaningfully have a say. This is particularly a problem given that many of these decisions won't get made in public by

politicians; they'll be made behind the scenes by people in the tech world as they create standards, informal norms, etc.

If we want more than a handful of people to shape the rules of the emerging tech road, we must ensure that as many people as possible in every community learn not only the technical skills but also the civic skills needed to truly participate. In Part 3, we will draw lessons from Citizenship Schools' experience teaching civic engagement, explore the implications of these lessons for emerging tech civic training, and provide a brief overview of some of the economic questions emerging tech will raise in the coming years.

One final note. While civic training must be rooted in specific values such as freedom, equality, and justice, it must also be designed so it can assist people from a wide variety of political ideologies and perspectives learn how to shape the direction of emerging tech, our communities, and our society. The goal of this training is not to push a specific political viewpoint, it's to help revitalize our democracy in an era of rapid technological and economic change.

3.2 WHAT CITIZENSHIP SCHOOLS CAN TEACH US

In the 1950s and early 60s, one of the major challenges the Civil Rights Movement faced in the Deep South was that voter suppression laws barred anyone from voting unless they could read and write. How could the movement help enough African-Americans become literate quickly enough to build political power -- especially in an environment where any efforts by African-Americans to win the right to vote might be violently suppressed? The solution: Citizenship Schools.

Citizenship Schools had a deep and profound impact on the Civil Rights Movement, which is why Dr Martin Luther King Jr called their creator, Septima Clark, the "mother of the Movement." Citizenship Schools taught their students the basic literacy needed to overcome voting restrictions. But they also taught them the civic literacy skills needed to win the struggle for voting rights and to understand how the political system worked so they could make their voices count.

A Community-Oriented Approach

Some of the strategies Citizenship Schools used were very similar to the best traditions of Extension Services:

Harness the Power of Community.

By being deeply embedded in communities, Citizenship Schools were able to leverage a community's strengths, including the bonds of friendship and support. For example:

- When recruiting a school's teachers, they targeted people who civil rights activist Dorothy Cotton described as "people with Ph.D. minds who never had the chance to get an education who were the natural leaders in their communities"
- Teachers often mobilized people in the community to help set up the physical space of the school, which could be located in the back room of a local store, a church, a beauty parlor, or other community institutions, which also allowed them to hide the school from local white elites
- Teachers used local social networks to recruit students, who might be leery of taking a class not only because of the fear of white reprisals but also because of the stigma of illiteracy
- When students' training was over, teachers encouraged them to recruit their friends, neighbors, and other people from their community to take the next class

Be Responsive to Local Needs, but Operate at Scale

Citizenship Schools were grounded in helping people develop more self-sufficiency in their daily lives and gain an understanding of how they could help change their local community. At the beginning of the first class, for example, teachers would ask students what they wanted to be able to do with reading and writing -- e.g., being able to read documents you had to sign -- which were then incorporated into the class. Classes also often discussed local community issues and how they might be addressed.

Although Citizenship Schools were designed to incorporate the unique circumstances and needs of an individual community, to play an important role in changing the South they had to operate at scale. According to historian J. Douglas Allen-Taylor:

The Citizenship School Movement trained more than 10,000 community leaders from 1957 to 1970 through nearly 1,000 grassroots, independent schools that operated at one time or another in every county in South Carolina, nearly 90 counties in Georgia, and in all of the heavily-Black areas of the rest of the Deep South. At one point in 1964, almost 200 schools operated simultaneously. Former Atlanta Mayor Andrew Young... said that the Citizenship Schools were the "foundation" of the civil rights movement, "as much responsible for transforming the South as anything anybody did."¹⁷

Civic Training in the Service of Justice

Although some Extension Services traditions embraced more moderate forms of civic engagement, Citizenship Schools were designed from the ground up to combine technical and civic skill training in order to fundamentally transform the economy and society, helping African-Americans win their freedom in the South.¹⁸

As we will discuss later in this chapter, Citizenship Schools faced a set of educational and political circumstances very different from our own. But we can still learn some useful lessons from some of the strategies they used to achieve their goals.

Blending Technical and Civic Training

Citizenship Schools didn't treat basic literacy and civic literacy as two separate subjects. They intertwined teaching reading and writing with understanding how the political system worked, from understanding their rights to learning the nuts and bolts of lobbying for better local services. The curriculum was designed to develop both the skills and the confidence to use them on behalf of themselves and their community.

This intertwining of basic and civic literacy occurred in every facet of the training. For example, in the Citizenship School workbook for students in Georgia, here's what it had to say about learning to write:

As you improve your writing, new worlds of pleasure will open and old fears will pass away. You will enjoy writing your friends. You will be able to write to your newspaper and express your views on the events of your community. You can write your Congressman or Senator to help him to vote for things that will help your people, and you will not be shy about filling out job application blanks, signing your name to your checks or registering to vote.¹⁹

Similarly, students would begin by learning to read simple words in the workbook and use them to write brief stories about their lives, then read more complicated stories about the lives of African American political heroes such as Crispus Attucks -- whose story included the discussion question, "how is the problem of [British] taxes like Negroes' problem of voting?" -- and Harriet Tubman. Other readings included passages on the political philosophy of nonviolence.

Learning by Doing, Developing Leaders

The workbook ended with a section on "Freedom Songs to Read and Sing." But right before that was a section entitled, "Planning a Voter Registration Campaign," which asked students for a substantially bigger commitment than singing:

A good citizen must be a registered voter. But the job does not stop there. We cannot rest until every citizen is a registered voter. You have been helped to register through this citizenship course. It is now your turn to help your neighbors. Plan a registration drive for your neighborhood or community:

This call to action was followed by a planning sheet, which include questions about a student's neighborhood such as:

- "What is the size of the Negro population?"
- "How many can we get to register?"
- "Number of Volunteer Workers needed to cover area"
- "Organizations to take part in the drive"

Next came a template to "Canvas your Neighborhood" as well as a list of "Suggested Steps for a Block Party" about voter registration, which included the following:

- "Plan a meeting for next week to give help to each other (if possible, arrange to start a Citizenship School)"
- "Have someone contact the persons who did not show up at the [block party] meeting."

There were 2 reasons for this approach. First, according to Clark:

The basic purpose of the citizenship schools is discovering and developing local community leaders. One of the unique practical features of the concept is the ability to adapt at once to specific situations and stay in the local picture only long enough to help in the development of local leaders. These are trained to carry on an ever growing program of community development. The secret stems from the emphasis and the reliance on local leadership. It is my belief that creative leadership is present in any community and only awaits discovery and development.²⁰

The other reason behind this approach was the belief that there is a limit to what you can learn in a classroom. Action in the real world is needed both to fully learn these lessons and to build the confidence to act.

Citizenship Schools use a similar approach for training its teachers. In addition to being required to take a 5 day workshop on how to run a Citizenship School, teachers were expected to research the hours and location of the voter registration office, election dates, the names of local politicians, and the location of the nearest Social Security office. Workshops ended with a discussion of how they would use what they learned to make a difference in their community when they got back home. Many people who became local civil rights leaders started out as being trained to run Citizenship Schools classes.

3.3 APPLYING THE LESSONS OF CITIZENSHIP SCHOOLS

We can't simply replicate what Citizenship Schools did. For example:

- Basic literacy could be taught in just one course. Emerging tech training almost always requires multiple courses, so it requires a different strategy -- e.g., if you incorporated civic education into every tech skills class, it would drive even the most civic minded students up the wall.
- Citizenship Schools were part of a large, well coordinated pre-existing movement, the Civil Rights Movement. In contrast, we are just at the beginning of a wave of tech civic engagement, so there is no pre-existing national, coordinated infrastructure whose goals and strategy can help guide each community's solution.

But the biggest difference between the circumstances facing Citizenship Schools and the ones we face is that Citizenship Schools were part of a movement that at that point in its history had a very clear adversary: defeat white supremacy and Jim Crow. That's a very different circumstance than the one we face as we attempt to make coding's economic benefits accessible to many people in every community. In fact, we're at a moment where there are many competing ideas and ideologies about what the real obstacles are and what we need to do to move forward.

So, how do we apply the lessons of Citizenship Schools to emerging tech education?

There's no simple answer. In fact, one of the central tenets of Citizenship Schools is that the best answers for serving the unique needs of a community come from the community itself. Outside experts and ideas from other communities can be extremely helpful -- it makes no sense not to take advantage of others' experience and knowledge. But there's no simple, one-size-fits-all solution that will work perfectly for every community.

So instead of offering a definitive answer, here are two thought experiments to help kickstart a conversation about the best way to combine technical and civic education.

Rebuilding the Training Road As They Walk It

In this example, a tech course and the surrounding ecosystem are designed so they help students learn how to "make the road as they walk it," engaging in rethinking the training/job pipeline as they go through it. Although it undoubtedly would also include some discussions of the issues raised in the next chapter, its focus is on giving students the nuts and bolts skills of a very simple form of civic engagement

At the beginning of a selected community tech class, the teacher would say to their students, throughout this course I want you to keep asking yourself the question, how can we improve this course for the next group of students?

Every few weeks, the teacher asks the students to briefly reflect on their experiences so far. For example, they might discuss which concepts they had the hardest time understanding. They might also brainstorm how to teach those concepts so they're easier to understand -- e.g., metaphors/examples from everyday life to clarify the concept, or a quick and dirty exercise to make it easier to understand. They might even split into pairs or small groups and briefly practice using one of the strategies they'd come up with. Then together they would write up their suggestions for future trainers.

At the end of the cohort's class, the teacher could lead them through an exercise where they discussed questions such as:

- How we can make this training better for the next group of students?
- Can we make it easier for people to get their feet wet before they take this class?
- How can the people who are developing the programming language and tools we are using make them easier for beginners?
- Are any of the changes we're proposing small and manageable enough that interested students could take on some or all of the work needed to make or advocate for these changes?

A similar approach could be used as students work their way through the rest of the training pipeline and continue to develop their skills, finding ways to help people reflect and act on their experience together.

Then every few trainings or once a year, the group or groups teaching the course might host a community dialogue and working session with the students, teachers, and others who have been involved. They could discuss what worked, what didn't, and how both the training and the ecosystem it is a part of might be improved. This could include:

- Tweaking the pace of the course
- Creating better on-ramps
- Holding a weekend hackathon in several communities across the country to brainstorm ideas about how to make the programming language/library taught in the class easier to use, concluding with plans to advocate for these changes with the tech community that's developing the language/library
- Discussing potential fixes for some issues that have come up with some apprenticeships and paid internships that several students participated in after taking the course
- Listening to feedback from students who recently got their first tech job to see if there are any issues they wish the class had better prepared them for, and brainstorming potential solutions
- Strategizing about how to get the resources to increase the number of classes and identify and develop more teachers from community

Finally, these efforts could serve as opportunities for students to get more involved in tech activism in the community -- helping to make some of the changes that had been discussed, becoming class teachers, or joining efforts to create the equivalent of Extension Services for coding in their community.

Telling Stories, Crafting Code

What if we interweave the art of creating stories in AR/VR, learning the craft of coding, and learning the beginnings of tech civic engagement skills? Here's an example of what it might look like.

Story circles are a strategy some community groups use as part of building community and exploring issues within a community. According to Roadside Theater, an Appalachian community group that pioneered the concept,

The stories we're able to tell ourselves and others, those we can understand and imagine, define not only what we believe to have already occurred, but what we believe to be possible in our individual and collective lives.²¹

The basic idea of a story circle is simple: a small group of people sit in a circle and share stories about their experience on a given topic or theme. But given that many people in communities written off by society aren't used to having their story and their voice taken seriously, this simple act can have a profound impact.

Here's how story circles might be intertwined with learning to code:

- 1) In the first workshop, a group of adults meet for a Friday night and half of a Saturday.
 - On Friday night, first they participate in a story circle, then they learn how to use one simple coding technique to try to very simply express their story in AR/VR. For example, each participant comes up with 3 words that sums up their story, then creates a VR page that displays their 3 words using a simple special effect that requires a tiny bit of coding -- e.g., having the 3 words fade in (exactly what they would do would depend on the tool and coding environment they were using).
 - On Saturday, they add a few tools to their storytelling toolbox -- e.g., code for adding a picture and allowing users to interact with it -- so they can create the first version of their story. The workshop alternates between a little instruction, a lot of playing and experimenting with code, and reflecting on the experience of using the tools and how they might be designed differently to make them easier to use for folks in their community.

- 2) The group continues to meet for a few shorter Saturday sessions that take place every other week.
 - Each time they learn one or two more coding techniques, a little more about VR/AR design and how to tell a story, and reflect about the experience so far. In doing so, they also build the trust and community they need to help them get over any fears, which is often half the battle.
 - In between these sessions, they work on their own or with coding buddies on their coding skill and their story
 - After the first few sessions, they would also begin to discuss how AR/VR could be used to help their community, what the future of emerging tech might look like, and how that future could be shaped so it benefits all communities (see the next chapter for some topics they might discuss).

3) Then they would start the second part of the course: learning the basics of creating very simple tools they can add to their toolkit. Depending on the coding library/framework they are using, before the course was taught some coders might need to add a library that made it a little easier to create very simple tools.

As they gradually gained some confidence around the idea of being tool makers as well as tool users, they would also be asked to start having discussions, brainstorming sessions, and story circles around questions such as:

- Imagine this system was designed for people you know who spent most of their lives working with their hands and feel uncomfortable or nervous about the idea of making a living from coding. Tell me a story about what it would be like to use this system if it had been designed from the ground up by everyday folks who were used to working with their hands?
- Suppose the people who built this system wanted home care workers to become augmented/virtual reality "power users." Tell me some stories about how they would use it and what their experience would be when they first got started, as they began to become skilled, etc.? Next, imagine there's a way they can make some money on the side through what they create. What might that look like? What are some values we might want to design it around? What are some issues you think such a market or system might need to overcome?

By the end of the class, those who want to continue should be able to start meeting on their own as a group (a.k.a. a Band of Brothers and Sisters). Their group could get and provide assistance through a network of other folks around the country who've gone through a similar experience and who -- with occasional help from world-class techies from around the globe -- have been helping to shape the path people take to keep improving their skills. Perhaps a few of them will decide to help teach the next set of workshops. And perhaps their group will join others in their community and in other communities in helping to begin building the equivalent of Extension Services for emerging tech and begin having conversations about how we might begin to shape the emerging tech economy so that every community would benefit.

3.4 HOW EMERGING TECH WILL TURN THE ECONOMY UPSIDE DOWN

The way our economy is organized and the rules by which it operates are based around a core set of assumptions. And over the next 20 years, some of those assumptions are going to be challenged or undermined by emerging tech. As a result, our society will face some very big decisions about the rules of the road for this new economy.

In this chapter, we'll discuss how emerging tech is likely to challenge some of the assumptions that underlie our economy and outline a strategy for thinking about our future while focusing on the problems we face today.

Robots, AI, and the Future of Labor

Without a working crystal ball, there's no way to know what impact robots/AI will have on work. For example, we have no way to know:

- How many jobs will be automated?
- Will robots and AI will destroy more jobs than they create?
 - Of the new jobs created by robots and AI, how many will be good paying jobs?

The one thing we can be pretty sure of is that robots and AI are going to break some of the assumptions that our economy has been based on.

For example, our economy is based on the assumption that when businesses grow, many people will benefit because of the jobs the businesses will create. But in the future, even if robots and AI don't create mass unemployment, there is a very real danger that many growing businesses won't create enough good paying jobs.

As a result, we need to start asking questions such as:

- How do we ensure everyone has financial security in a world where people may not be able to depend on having work?
- Should people whose work is being automated away have some say in the process, and if so, how?
- Should we try to create incentives so we automate away much of the work that no one wants to do?
- As a society, do we want to ensure that the enormous profits and wealth that are being produced by automation help everyone? For example:
 - Should we try to move towards an economy where most people don't have to work full time unless they want to?
 - Do we want to try to give everyone a real opportunity to express their creativity and explore their full potential?

For most people, these questions don't have simple or easy answers. That's why it's crucial that we start creating space in our society to begin exploring them. As the Gershenfeld brothers

argue at the beginning of Part 3, if we wait too long before asking these questions, it may be too late for our answers to matter.

Creative Works

In an economy increasingly dominated by emerging tech, often the greatest economic value won't come from physical objects but from the creative works that power them:

- A robot's operating system and its "apps" that let it cook food
- A recipe that tells a robot how to make an apple pie
- A virtual pet in augmented reality
- Code that can create different types of forests in virtual reality
- A design for digitally fabricating a comfy chair
- The patent for sensors that allow a robot to etch innovative patterns on glass -- and the design and code that lets anyone digitally fabricate that sensor

Unlike physical objects, the cost of making a copy of a digital creative work is close to zero.

We can already get a glimpse of the potential of an economy where digital creative works are central and easily available:

- If you want to learn a new skill, odds are there dozens of tutorials freely available on YouTube.
- Many recent AI breakthroughs have been driven by techniques known as "deep learning," which spread rapidly because both the ideas behind it and the open source programming libraries for implementing it are freely available.

Now imagine a world in which millions of people in communities across the globe are contributing to a body of emerging tech that is accessible to anyone. The potential of that pool of creative works is staggering.

But in a world where digital creative works are either freely or cheaply available, how do its creators make a living? If digital creative works are increasingly central to our economy, the experience of many musicians and newspaper reporters today may be our canary in the digital coal mine.

In short, one of the central economic questions we will need to grapple with in coming years is, how do we reward creative work while ensuring that the creative bounty it generates is widely shared?

Answering this question will be important regardless of the impact AI and robots have on employment. Even if AI and robotics destroy more jobs than they create, there will still be an abundance of opportunities for making creative works that many people value. The big question is how the new rules of the road will shape who benefits.

Digital Fabrication

As we discussed in Part 3's introduction, MIT's ambitious plans for digital fabrication are just beginning to take off. So far, they've created a global network of over 1,000 community-based "Fab Labs," filled with 3D printers, laser cutters, and other fabrication and electronics tools, as well as people who can teach you how to make an amazing array of objects. Fab Labs all use the same tools and processes, so if someone in the global network creates a design for a beautiful chair they build in their Fab Lab, anyone else in the network can also make it.

MIT's Center for Bits and Atoms believes that in the coming decades, between conducting primary research and testing out some of its ideas through the Fab Lab network, it will be able to accomplish a dramatic technological transformation:

From machines in a Fab Lab that make things,
to machines that make parts of machines,
to machines that self-reproduce,
to building with digital materials,
to materials that are programmable and can turn themselves into parts.²²

In short, they are confident that by the end of journey, they will create a personal version of something like Star Trek's replicators.

Even if they're only partially successful, this transformation could have a wide range of effects. As some members of the Fab Labs movement like to ask, what if instead of having to take a job you don't like to make money to buy an object, you could just make the object yourself? Even if it's only true of some objects, it raises deep and profound questions about how we shape the rules of this new economy so everyone benefits.

As a result of those questions, spearheaded by Barcelona, there's a growing network of "Fab Cities" dedicated to experimenting together to see if they can leverage Fab Labs to build a more just, democratic and sustainable future. Their goal: by 2054, to move

from 'Products In Trash Out' (PITO) to 'Data In Data Out' (DIDO). This means that more production occurs inside the city, along with recycling materials and meeting local needs through local inventiveness. A city's imports and exports would mostly be found in the form of data (information, knowledge, design, code)...

We need to reinvent our cities and their relationship to people and nature by re-localising production so that cities are generative rather than extractive, restorative rather than destructive, and empowering rather than alienating, where prosperity flourishes, and people have purposeful, meaningful work that they enjoy, that enables them to use their passion and talent.²³

It's anyone's guess as to whether this movement will come anywhere close to achieving its ambitious goals. But regardless of what you think of their approach, one thing is certain: the questions they're raising and the scale of answers they are proposing is exactly the kind of work communities across the globe need to start engaging in if we are to take advantage of these potential opportunities.

Thinking Ahead While Focusing on Today

Given the impact emerging tech will eventually have on the economy, we need to begin raising the kinds of questions laid out in this chapter. We also need to start asking broader questions such as:

- What role should work play in our society?
- What are our needs as human beings?
- What are our deepest values, and how can our economy help support them?

But there's a limit to how helpful it is to wrestle with these questions today. The most profound impacts of emerging tech probably won't affect us until two decades from now. In the meantime, given that so many communities are currently suffering, we need to spend the bulk of our energy on solving problems in the here and now.

Moreover, even if we could come up with detailed answers today as to how we want to influence the impact of emerging tech on our future economy, we couldn't build a realistic roadmap to get to where we think we want to go. The endpoint is too far away to have any confidence as to how we'll get there.

But if we do it right, it's still extremely useful to think and debate about the future. Perhaps the best way to manage the tension between today's immediate needs and tomorrow's potential is to keep asking:

- How can we use our answers to these long-term questions to shape today's plans?
- How can we design today's plans so they open up space and create opportunities to build the future we want?
- How can we use our work in the next few years to test out the ideas and assumptions we have about the future we think we want to build?

It is a rare and precious gift to have a glimpse of the dangers and opportunities we'll face over the next 20 years. Let's not squander it.

Conclusion

Getting Started

If you agree with some or all of the vision laid out in this report and you want to try it out in the real world, where do you begin? Start from where you are.

Social change is a team sport; if you're going to succeed, you'll need to form a group. Perhaps you could start with people you know from church, your union, where you work, places you volunteer, or your friends. If you have the kind of contacts to do something more ambitious such as convening people from across your community, go for it -- but it isn't necessary to get started.

Then begin having discussions, asking questions, figuring out what you agree on and don't, what you know and don't know, and educating yourselves. And start small, but keep your eye on the prize. Bringing about a transformation on the scale that's necessary is about a dance between big goals and little steps, about dreaming big without ever letting your group's ambitious goals overwhelm you.

When you first get started, there are three steps your group will need to take: envision, map out the terrain, and start learning by doing. Odds are you'll end up cycling through these steps more than once.

1) Envision

- What would your community or region look like in 10 or 20 years if you succeeded? What would it feel like?
- Roughly how many people in your community would need to work in emerging tech to improve the overall economic opportunities in your community? And how would you know that enough people had found a home in emerging tech to make a real difference in your community or region? This is a crucial step. Too many people who want to democratize tech will count how many individuals they're reaching but will never take a hard look at whether they are having a major impact on specific communities.

2) Map

- What groups or individuals are already attempting to make tech more accessible in your community? For example: community groups, tech activist meetups, colleges and universities, vocational education efforts, sympathetic local or state politicians, people inside tech companies who care about empowering people with tech. Are there any existing efforts your group should join?
- If you're from a thriving middle-class community, are there nearby communities who are looking for help? If there are opportunities to help out, it's extremely important to enter into these communities with respect and more than a little humility; from Harlem to Harlan County, nobody likes well-meaning but patronizing outsiders.
- What else can you plug into nationally -- e.g., churches, unions, and other national networks of civic organizations?
- Does your community have enough resources or the right resources to help democratize emerging tech? For example, what kind of support do the groups who are already trying to democratize tech have and what do they need?

- How inclusive are your efforts? And if they aren't inclusive enough, are there networks in your community you could partner with in your efforts to become more inclusive?
- What major obstacles does your community face, such as lack of broadband access or literacy issues? When and how can your group attempt to make some progress overcoming these obstacles without bogging down the entire project?

3) Try

- What is the smallest step your group can take to get your feet wet and start testing out your ideas and assumptions?
- Once your group has gotten your feet wet, how can you keep taking incremental steps that move you forward without feeling overwhelmed?
- How do you get comfortable with the fact that this process inevitably involves trial and error? How does your group build a culture of being open and honest about your mistakes so you can learn from them?

Some Initial Issues You May Encounter

- **Don't Sweat What You Don't Know, Ask for Help.** As soon as you start this journey, you're going to run into issues that you don't know how to address. That's perfectly normal; just track down some help. For example:
 - **Job Stats.** For the question of how many people need to end up getting jobs to make enough of a difference in a community, odds are there are people who work for your state who are responsible for making projections of what the "workforce" of your state will look like in the future. There are also probably academics who have wrestled with this issue. And there may be some national policy shops who have experts who would be happy to come up with a rough estimate. To find and connect with them, start by searching online or asking librarians at your local library.
 - **Emerging Tech Coding.** If no one in your initial group is fluent with emerging tech, odds are there are people in your community who are experts and who would be happy to help you figure out how to get started. Just make sure to follow the example in Part 3 and use your group's experience learning the tech to ask, what would the tech look like if it were designed from the ground up to be accessible for people like you?
 - **Civic Engagement.** Many of the ideas in this report will be familiar to community organizers and others who are experts in community-based strategies (including some extension agents). If your group doesn't include anyone who has these civic skills, there are certainly people in your community who do and would be happy to share them. As your group begins to learn the basics of civic engagement, you should think about how these ideas and skills could be incorporated into tech trainings.
- **Community Networks.** If you hope to help nearby communities, odds are there are people you can connect with in your community who may not be knowledgeable about tech but who are knowledgeable about those communities' networks and have some contacts in those communities.
- **Focus on Diversity From the Jump.** One of the painful lessons of tech is that if a starting group is mostly white, male, and middle class, odds are it's going to stay that way as it grows. If your group has diversity issues, it's critical that you focus on becoming more inclusive from the very beginning.

Setting Goals Without Setting Ourselves Up

Setting goals is critical to success -- especially when you're dealing with a problem where the solution may require orders of magnitude more resources than are currently engaged. Your group also needs goals so you have a very rough idea of how far you are along the path to success.

But it's easy to get tripped up by goals. A few thoughts on how to set goals that make your work easier rather than harder:

- **At the very Beginning, Numbers Matter Less.** Don't focus too much on numbers at the very beginning; otherwise you'll end up feeling so overwhelmed you give up. Realistically, you've probably got 8-10 years to hit your most ambitious goals. The point of asking these questions now isn't so you'll worry about nailing your numbers right away, it's to ensure you take advantage of the luxury of having that much time.
- **How Many Zeros?** Similarly, exact numbers aren't important early on. What you need to know is, are enough people getting trained and either getting good paying jobs or creating small businesses to boost your community's economy, or does your community need to train 10 times or 100 times as many people?
- **Track Diversity.** As the tech world has demonstrated, if organizations don't track their diversity they aren't likely to improve it. Once you're starting to make progress, make sure you think through how you will break down your goals by race, gender, income, urban vs. rural, etc. to ensure that everyone in your region will have a fair shot at jobs and co-ops/small business opportunities in emerging tech.
- **Don't Juke the Stats.** As soon as stats are treated like grades, institutions will get creative in figuring out how to manipulate the numbers so it looks like they are succeeding -- what the TV show *The Wire* called "juking the stats." So if politicians or funders start hammering on exactly how many jobs should be created and setting unrealistic expectations, push back hard.

Expanding Your Efforts

Once your group has made some progress, you'll need an iterative approach that helps you stay on course:

- **Expand the Circle.** As your project begins to rack up some small successes, what other parts of the community do you want to invite to become partners? Or you may do the reverse -- for example, develop some proofs of concept that will help others wrap their heads around what you are trying to accomplish, then bring in more facets of the community and encourage them to assume leadership roles.
- **Stay Inclusive.** How do you ensure that as you grow all facets of your community are represented?
- **Develop Leaders.** As Part 3's overview of Citizenship Schools demonstrated, identifying and training new leaders is a crucial part of building a vibrant grassroots movement. As you grow, you'll need to start incorporating leadership development into your plans if you haven't already.

Scaling Up

As crucial as local action is, at some point efforts in individual communities won't be enough. If we want to ensure that no communities will be left behind and that every community will have the resources it needs to succeed, we will need to develop something similar to Extension Services for emerging tech.

We can't know in advance what the right solution will be -- for example, whether it can be funded primarily through private means, a balance of public and private, or through largely public means as was required for agriculture. The only way we'll know what works and what doesn't is through experimentation. But regardless of the details of the solution, one thing is clear: we will need a solution that's up to the scale of the problem.

To create the equivalent of Extension Services, we will also need to build a national network for civic action -- a project that Makers All, which sponsored this report, hopes to facilitate. Part of what such an effort will require is building connections between community experiments and creating dialogues across communities so we can organically develop the solution. But it will also require that communities work together to bring pressure on government, large tech companies, and other large institutions so they mobilize the resources necessary to ensure every community shares in the opportunities and abundance created by emerging tech.

As we attempt to mobilize these resources, one of the advantages we have over some other civic efforts is that we may be able to mobilize the self-interest of one of the biggest players: the tech world. Although there will undoubtedly be resistance in some parts of the tech world to truly democratizing emerging tech, in the long run it's a no-brainer:

- If we succeed, we will greatly expand both the pool of talented people and the market opportunities in emerging tech. Big tech companies will have a smaller slice of the pie, but the pie will be much, much bigger.
- And if we don't succeed? What's behind Door Number 2 is too scary to contemplate, but it'll undoubtedly include some version of "peasants with pitchforks" bent on destroying the Frankenstein of robots/AI they see as a threat to their community.

The need for bold action isn't news to anyone in the tech world -- that's why discussion of radical ideas like Universal Basic Income are now commonplace. The issue isn't whether we need to act, it's what the right action is. This shared understanding in tech doesn't mean action on the scale we need is inevitable, but it greatly improves the odds of success.

Special Considerations

The Role of Tech Companies

Although tech companies will need to be involved in community-wide efforts, they also have a distinct role to play. If you work for a tech company that wants to make the tech you develop more accessible, here's what you and other people at your company can do.

Building Relationships to Develop Community-Oriented UX

The first thing you'll want to do is to start exploring how to implement community-oriented UX.

- If you work in a large company, odds are there already community groups that your company has a relationship with, such as community groups your company has given grants or donations or groups where some of your staff volunteer. Using these pre-existing relationships, you can see if these groups might be interested in building a partnership.
- There's a good chance your community partners aren't familiar with community-oriented UX, so you'll need to explain what you're trying to do and why. But if the group has trainers who are skilled at training folks in the community to use tech, there's a good chance that once they understand what you're trying to do and why, they'll be interested in at least trying a short experiment.

There are 2 keys to making this partnership work:

- **Pick the Right Tech.** If you have more than one framework/library/etc. to choose from, pick the tech that's easier to make accessible. Right now, for example, if you're developing tools in augmented and virtual reality, these are a pretty safe bet.
- **Find The Right Partner.** It may be that the community groups you already work with aren't ready to make the leap. Or it may be that they aren't a good fit right now -- for example, they just don't have the time given their other priorities. There may be other community groups you haven't worked with who make more sense as partners -- perhaps even groups that haven't traditionally engaged in tech training but have a lot of capacity and experience training in other areas.

Once you've figured out the right Tech and the right partner(s), you want to start your collaboration with a pilot project. The goal of the pilot is to get your feet wet and start learning how to do community-oriented UX in a way that works for everyone involved. If you're ready, it also might be worth considering using these first baby steps as a chance to begin conversations about the possibility of hiring one or more community groups as community-oriented UX consultants once they've gained some expertise.

Small Startups

For small startups, the situation is more complicated. Most startups don't have the kind of resources that a larger shop has -- and there's a good chance they don't have any relationships in the community to start from. But it's still worth exploring to see if there are ways you can build a partnership.

Be Flexible

All of the above is meant only as a guide; you'll need to adapt it to your circumstances. For example, it may take some time before you'll figure out the right community partners. In the meantime, there's no reason you couldn't start working on smoothing the learning curve. In the experience of this report's author, many emerging tech frameworks, etc. have glaring UX shortfalls -- issues where anyone with experience teaching coding to non-techie adults could point out a number of problems that could be fixed even before you start using formal UX techniques.

Using Economic Leverage To Help Others

Although making the wealth of emerging tech accessible in every community is a critical part of confronting the robots/AI employment threat, it isn't the only issue we'll need to address. One pressing concern is that even if huge numbers of jobs aren't eliminated, too many of the remaining jobs may not pay well. For example:

- **The Paradox of Automation's Last Mile.** Researchers Mary Gray and Siddharth Suri have shown that for every round where AI automates away work, humans play a critical role in developing or cleaning up the data needed to train AI and handling the behind-the-scenes work that AI can't yet manage (e.g., responding to a customer complaint).²⁴ Most of this work can be done by just about anyone, so it rarely pays well.
- **Care Work.** It's not clear if robots will be able to take care of children or the elderly, but even if they could, we might not want them to. But care work has never paid well because it's been considered "women's work."

Given that people who obtain full-time jobs or create small businesses/co-ops in emerging tech will be in an economically advantageous position, over the long run we will need to see if there are ways they can use their position to help people who end up in lower paying jobs. Sometimes this may simply be a matter of supporting their organizing efforts. Other times it may involve helping them to bring the economic benefits of emerging tech into their jobs. For example:

- Pushing companies to develop the emerging tech home care aides use so that it enables some home care aides to become power users, gaining more skill and making it easier to advocate for better pay
- Helping organized nursing home aides gain a say on the direction of efforts to automate their work so their jobs become less physically demanding and more mentally and emotionally fulfilling over time

It's unrealistic to expect this kind of support early on -- emerging tech activists will already have their hands full. But as they dream about when their efforts could lead 20 years from now, it's worth at least beginning to ask questions about how successes with emerging tech might be leveraged to help all people in their community.

Forging Bonds Between Communities

Although this report is focused on using the opportunities of emerging tech to help communities heal themselves, this work might also provide an opening to heal the divides across communities.

We live in an era where much of our politics are polarized -- and in a democracy that's necessary at times. But in such a polarized time, we also need ways to rebuild the bonds between us.

There are few better ways of reforging our bonds than in the crucible of working together. Working together and learning from one another is an efficient and effective means of operating. But if our work is structured properly, it can also build connections among people and communities that have a deep and profound effect on how we see others and how we see ourselves.

In short, as we work together, helping one another as we struggle for a better future for all of our communities, black and white, rural and urban, red state and blue, we may also find our way to rediscovering our common humanity.

Notes

¹ Pablo Illanes, Susan Lund, Mona Mourshed, Scott Rutherford, and Magnus Tyreman, "Retraining and Reskilling Workers In The Age Of Automation," McKinsey Global Institute, January 28, <https://www.mckinsey.com/global-themes/future-of-organizations-and-work/retraining-and-reskilling-workers-in-the-age-of-automation>

² In the US, Cooperative Extension Services has been known by a wide range of names, including Agricultural Extension Services, Extension Services, and Extension. For the purpose of this report, we will use either Cooperative Extension Services or Extension Services.

³ Ruby Green Smith, *The People's Colleges: A History of the New York State Extension Service in Cornell University and the State, 1876-1948* (Cornell: Cornell University Press, 1949), pp. xxxi-xxxii.

⁴ Bill Gates and Melinda Gates, "Our Big Bet For The Future: 2015 Gates Annual Letter," <https://www.gatesnotes.com/2015-Annual-Letter> (if your browser has difficulty reading it, see <https://www.ozy.com/opinion/bill-gates-where-the-third-world-gets-health-right/39034>).

⁵ Civic engagement is an important part of some Extension Service traditions, but the author of this report believes it's accurate to say that it was rare for their brand of civic engagement to go as deeply as Citizenship Schools did. Citizenship Schools were attempting to fundamentally change how the economy and society was ordered, changing the balance of power so African-Americans would become equal citizens. If Extension Services had been operating in an equally deep level, it's hard to see, for example, how it would have turned from supporting small farmers to focusing on Big Ag without sparking a national, highly organized resistance by the small farmers it was beginning to abandon.

⁶ "Building an augmented reality (AR) application using the WebXR Device API," <https://codelabs.developers.google.com/codelabs/ar-with-webxr/#5>

⁷ "The Definition of User Experience (UX)," <https://www.nngroup.com/articles/definition-user-experience/>

⁸ "Usability 101: Introduction to Usability," <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

⁹ Anil Dash , "The Blue Collar Coder," <http://anildash.com/2012/10/the-blue-collar-coder.html>

¹⁰ Andrew Ko, "Stanford Seminar - Learning to Code: Why We Fail, How We Flourish," <https://www.youtube.com/watch?v=mkzHIhKaUX4>

¹¹For a great overview of why fast.ai decided to build a library on top of Pytorch, see Jeremy Howard, "Introducing Pytorch for fast.ai," fast.ai blog, September 8, 2017, <http://www.fast.ai/2017/09/08/introducing-pytorch-for-fastai/>

¹²Andy J Ko, "We Need to Learn How to Teach Machine Learning," <https://medium.com/bits-and-behavior/we-need-to-learn-how-to-teach-machine-learning-acc78bac3ff8>

¹³ Aaron E. Carroll, "What Barbershops Can Teach About Delivering Health Care," New York Times, May 21, 2018, <https://www.nytimes.com/2018/05/21/upshot/what-barbershops-can-teach-about-delivering-health-care.html>

¹⁴Jay Hanlon, "Stack Overflow Isn't Very Welcoming. It's Time for That to Change," Stack Overflow blog, April 26, 2018, <https://stackoverflow.blog/2018/04/26/stack-overflow-isnt-very-welcoming-its-time-for-that-to-change/>

¹⁵ "A Conversation with Neil Gershenfeld," Edge, January 23, 2015, https://www.edge.org/conversation/neil_gershenfeld-digital-reality

¹⁶ Neil Gershenfeld, Alan Gershenfeld, and Joel Cutcher-Gershenfeld, "Introduction," *Designing Reality: How to Survive and Thrive in the 3rd Digital Age* (Basic Books: 2017), <http://designingreality.org>

¹⁷ J. Douglas Allen-Taylor, "Septima Clark: Teacher to a Movement," <http://www.safero.org/articles/septima.html>

¹⁸ For more on the distinction between Extension Services and Citizenship Schools, see the Introduction's last endnote.

¹⁹ Southern Christian Leadership Council conference workbook,

https://www.crmvet.org/docs/cit_schools_workbook.pdf

²⁰ Septima P. Clark, "Literacy and Liberation," *Freedomways*, 1st Quarter, 1964,

<https://www.crmvet.org/info/cs.htm>

²¹ Roadside Theater, "About: Story Circles," <https://roadside.org/asset/about-story-circles>

²² Fab Foundation, "What Is a Fab Lab?" <http://www.fabfoundation.org/index.php/what-is-a-fab-lab/index.html>

²³ Tomas Diez, "Fab City Whitepaper: Locally Productive, Globally Connected Self-Sufficient Cities,"

<https://fab.city/documents/whitepaper.pdf>

²⁴ Mary Gray and Siddharth Suri, "The Humans Working Behind the AI Curtain," *Harvard Business Review*, January 9, 2017, <https://hbr.org/2017/01/the-humans-working-behind-the-ai-curtain>. They also have a great book coming out in May 2019 about this subject, called *Ghost Work*.