67. *Mass Production*

Mass production is scaling's handmaiden. Solving bigger or harder problems in software isn't a matter of working harder or of adding more people, but of working smarter. Adding more people rarely increases quality but instead just creates a glut of output. To me, this explains Jeff Sutherland's complaint that 60% of software features aren't used: enterprises work from an overstaffed push model instead of a market-driven pull model. Or mass production can rob the market of choice: Taichi Ōno notes that America cut automobile production costs by mass-producing fewer car models — an idea he dismisses as unsound.

Scaling for mass production reduces prospects for quick, piecemeal delivery and agility in the product cycle. A small team produces products in the way that Christopher Alexander of patterns fame advocates evolution through local adaptation and piecemeal growth. Small increments allow for the dutiful incorporation of feedback along the way. Adding people adds more functionality — as well as the overhead of coordinating each of those functional contributions — adding new ones faster than the existent ones can be fine-tuned. And as we noted earlier, staffing up speeds delivery only for work that can be partitioned, as is rarely true for complex systems. The same is true for process as for product. Great ships turn slowly, and large organizations struggle to be agile.

The end of work force growth is hardly the end of progress. A system can always grow in scope and richness over time, in step with the market's ability to absorb new features and to steer product content with timely feedback. However, the best way to grow isn't by re-inventing the wheel yourself, but instead by standing on others' shoulders.

Partnerships are a cornerstone of lean. Much of the Japanese economy functions through federations of cooperative ecosystems. Toyota in many ways isn't just a company, but a much broader federation of cooperating entities. Even brands that appear as single entities from the outside — like Apple — work more as an ecosystem of cooperating parts with independent paths to market.

If you want to think big, don't think scaling, but instead consider leveraging your technology by making it public — as Tesla has licensed its charging station technology. Open Source works much in this way. Yes, that is certainly a scaling of the work force, but it's a work force that's explicitly not a work force. It's another development paradigm. When people ask me; "What comes after agile?" that's where I point.

Other consultants instead offer to come in and help you scale up your Scrum.

Individual firms and their components are the base of a product economy, feeding higher-level ecosystems that emerge as interactions between them. Firms like Apple are conscious to create an ecosystem, which they do by balancing a strategic overview with offerings that each have an enterprise identity in their own right. Scaling happens by adding new value streams for iTunes or phones or computers or software. Value-adding accessories emerge from other firms — not just by producing more phones.

And if you're just starting out, raise the bar even further. Imagine a business where efficiency was turned into free time instead of increased output. Set a business pace based on lean principles instead of excess. Have the team deliver "enough" and to spend the rest of the time at the beach.

Excessive production, and the resulting waste in discarded packaging, uneaten food, and overuse of wasteful and inefficient energy sources that fuel overproduction, are killing our planet. We learned two decades ago to stop counting lines of code. Now let's stop counting noses.