Why go Agile?

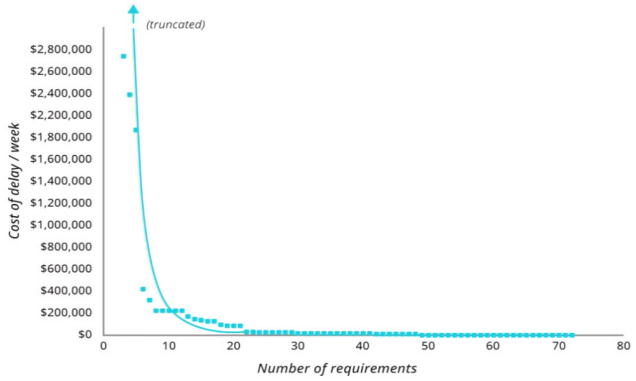
Truth will sooner come out of error than from confusion.

W/F Code Waste

- In a study of 400 waterfall projects:
  - Only 5% - 15% of the code was ever used.

Prioritize

batching up work



"Black Swan Farming using Cost of Delay" | Joshua J. Arnold and Özlem Yüce | bit.ly/black-swan-farming

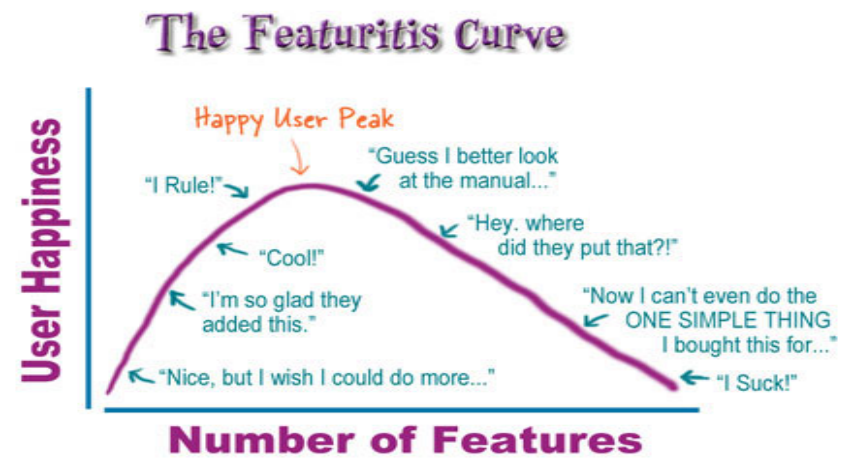
One company did cost/benefit analysis of its requirements.

3 requirements were costing the company millions of dollars per week of delay.

Most requirements were just good to have.

So everyone in the company should be working on these 3 requirements, as far as possible.

Agile Requirements



I have always wished that my computer would be as easy to use as my (landline) telephone. My wish has come true. I no longer know how to use my (mobile) telephone – Bjarne Stroustrup.

Why choose Agile?

- No Yagni.

Agile Methods

- Promote rapid delivery of value to customers
- Provide timely and regular visibility of the solution to customers, product owners and stakeholders.
- Deliver increases in productivity, quality and ROI for software development organizations

Minimum Process. Maximum Value.

Deliver benefits early

Avoid significant rework by only doing just-in-time detailed design

Raise quality by moving testing forward in the process

Become responsive by supporting scope adjustments every iteration

Become reliable by instituting regular heartbeats to the team

Increase estimating accuracy by working in small chunks

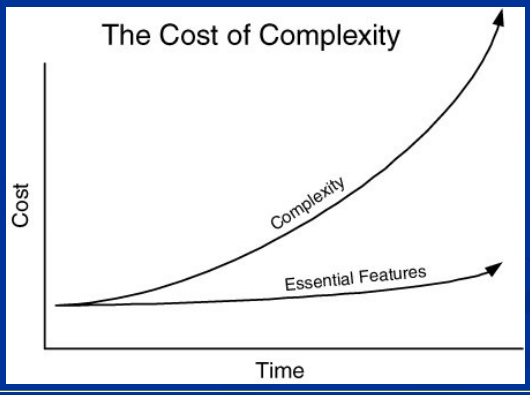
Decrease risk by always having working software

Increase throughput via real-time visibility

Increase team moral by dropping the “death marches.”

The Enemy is complexity

- Complexity is like cholesterol: it clogs up the arteries of an organization.



Tame Complexity:

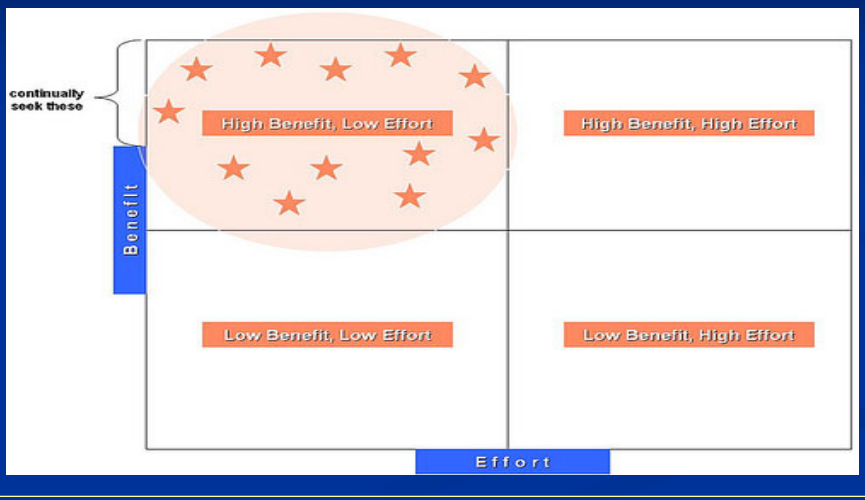
Aggressively limit the features and functions that make it into the code base

> For every 25 percent increase in problem complexity, there is a 100 percent increase in complexity of the software solution.

The code base should be continuously refined to better meet the customer needs. If that is not clear, the programmers must not write a line of code.

> Every LOC costs money to write and more money to support. It is better that developers be surfing than writing code that won't be needed.

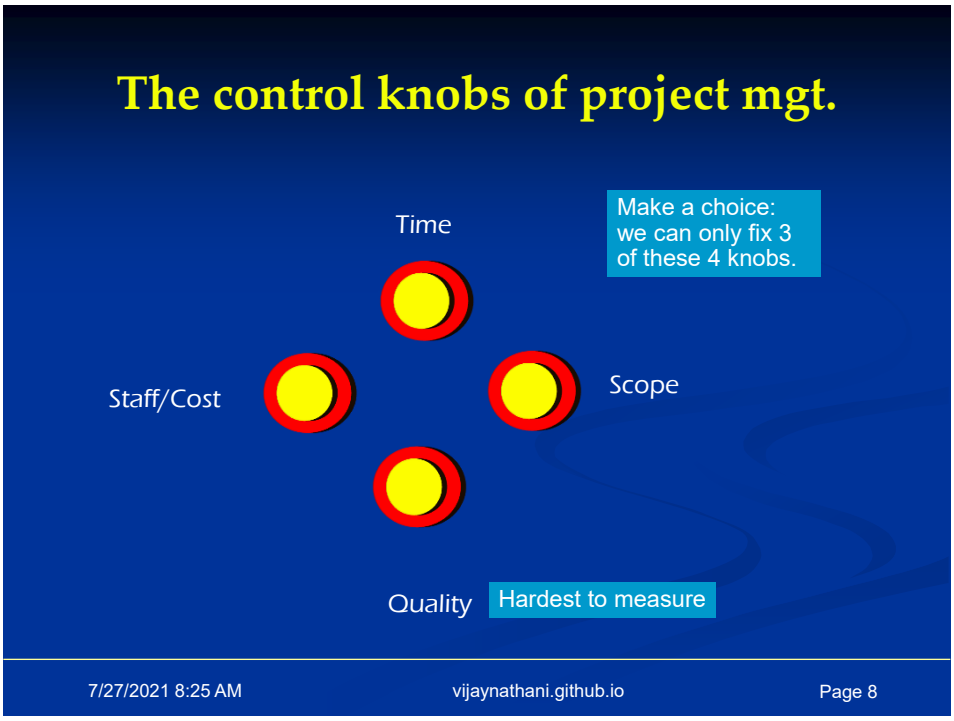
Agile aim



Any fool can make things bigger, more complex, and more violent. It takes a touch of genius—and a lot of courage—to move in the opposite direction.

Never do anything that is a waste of time and be prepared to wage long, tedious wars over this principle

An 80 percent solution today is better than a 100 percent solution tomorrow.



It takes less time to do a thing right than to explain why you did it wrong.

Small maladies, as doctors say, at their beginning are easy to cure but difficult to recognize…but in the course of time when they have not at first been recognized and treated, become easy to recognize but difficult to cure.

Define Quality: quality is what you do so that your velocity working in the code base does not degrade - you don't really know how to get it, but you can tell when it's gone. It's the great mystery of software. But we spend a lot of effort trying to get and keep it...

- The SW doesn't do something that the product specification says it should do
- The software does something that the product specification says it shouldn't do
- The software does something that the product specification doesn't mention
- The software doesn't do something that the product specification doesn't mention, but should
- The software is difficult to understand, hard to use, slow, or will be viewed by the end user as not right

Time vs. Bugs

Quality Is free, but only to those who are willing to pay heavily for it.

High quality = High Productivity in long Run

Why?

Less rework.

A policy of "Quality—If Time Permits" will assure that no quality at all sneaks into the product.

==========

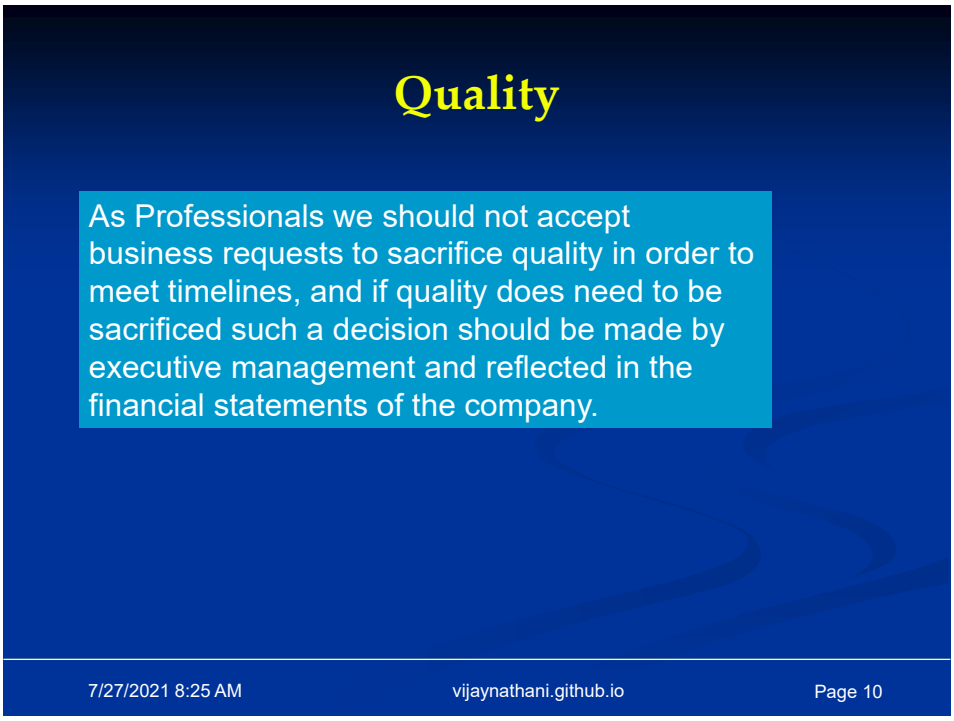
Quality is a collection of attributes.

Errors tend to cluster.

Efficiency stems more from good design than from good coding.

The unavoidable price of reliability is simplicity.

Measuring compliance to process doesn’t measure outcome. It is better to measure the outcome than process.



Not compromising on quality is not only your professional obligation but it is also important for your own joy of work and is critical for the company.

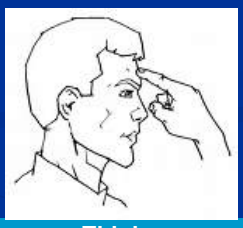
Companies that choose to cut quality in order to speed up time to market / competitiveness. The problems with doing this is that it reduces team velocity on future iterations, eventually companies can back themselves up into a corner and velocity can be negligible.

Quality is internal and external. External quality is what the customer sees. Internal quality is no duplication, readable code, etc.

Customer only see the tip of the software, the part they use.

Software developers focus on the unseen elements, but still need to consider how it will look to customers.

Speed



Think:
We don't have time
OR
We have too much to do

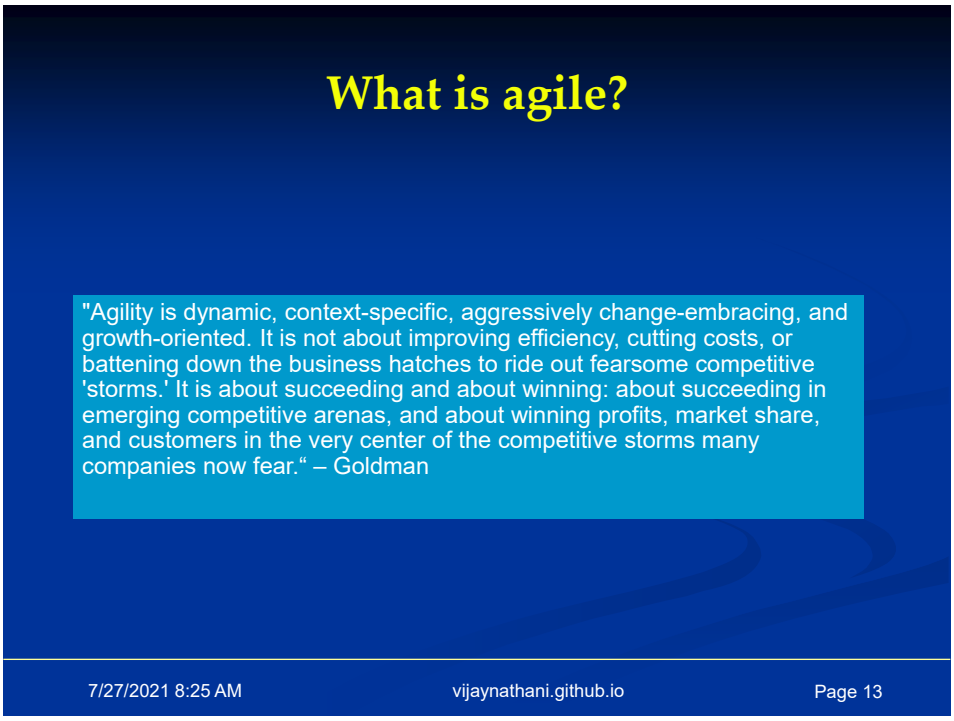
Don't do this

It is easy for programmers to feel guilty when the customer is disappointed, and for that guilt to turn into shorter estimates, not for any technical reason, but because they want the customer to be happy.

Usually the speed is at the cost of quality.

The developers are probably cutting down on testing / refactoring / both.

The team will probably end of spending much more on debugging later on.



Fixed prices are broken promises.

The only thing constant in life is change.

ThoughtWorks, Inc., a transnational IT services firm, released a commissioned study by Forrester Research that examined the overall economic impact of ThoughtWorks agile development approach. The study found agile techniques reduced cost, risk, and time-to-benefit while improving overall quality on complex software development projects.

Forrester, a leading independent global technology advisory research firm, interviewed four ThoughtWorks clients for the study. Participants indicated they realized or expected to realize 29 to 66 percent risk-adjusted return on investment over three years using ThoughtWorks agile development approach.

Analysis of a composite organization, constructed by Forrester based on client interviews, found an investment in ThoughtWorks agile development approach:

•Reduced cost by 57 percent compared to other IT solutions for similar complex projects.

•Reduced effort by 62 percent compared to alternatives, including in-house development and previously employed consultants.

•Reduced critical defects by nearly 80 percent.

•Reduced overall defects by more than 60 percent.

"This study points to one of the primary reasons we've been able to more than double our size during the absolute worst period in history for technology services," said Roy Singham, ThoughtWorks founder and CEO. "ThoughtWorks was built on the premise that the single most important success factor on complex projects is people. Now the Forrester study findings confirm that our people, coupled with our agile approach, differentiates us on complex, highly strategic software projects. Over the last several years in particular, Global 1000 companies have come to place a high value on these capabilities.“

According to the study, clients, regardless of the type of ThoughtWorks engagement, saw two key value propositions:

1. The quality of the ThoughtWorks staff and their ability to handle difficult projects drove efficiency within the organization, reducing the potential for defects and rework down the road. Improved visibility into project progress as well as use of automated tools such as Cruise Control further minimized unforeseen defects.
2. The use of an agile approach, requiring close communication between IT and business stakeholders, delivered key business requirements, and thus benefits, sooner. Business users had control from the start of the project, ensuring that key business requirements would be met. Several customers noted that prioritizing benefits upfront was a key component in seeing the value of ThoughtWorks solution.