Software Development Challenges

Hard to develop?

- Why is it hard to develop software within budget and time?

- Quality of code

- When you say it’s done, what do you mean?

- How good is the code you write?
Risk in development

- You write code based on what you know
- When was the last time you had to change the design?
- What happened after you changed it?
- Does your code turn into a loose cannon towards the deadline?

Efforts to minimize Risk

- Change in inevitable
- You don’t want to wonder what the effect of a change is
- Feedback is critical
- Frequent feedback is necessary
- You want to know right away if you broke the code, isn’t it?
Software Development

- What’s software development like?
- We often get compared to other human endeavors
- Let’s study some of those
  - Bridge Construction
  - Medicine
  - Flying

Bridge Construction

- Safety Concerns
- Strong metrics and standards
- Often construction and design are separated
- Innovation and construction are separated
Medicine

- “Health was thought to be restored by purging, starving, vomiting or bloodletting”
- Both surgeons and barbers were involved
- Rate of infection was high before Joseph Lister introduced Germ theory
- As human, we learn from our mistakes
  - We reject ideas
  - We take time
  - We learn eventually

Flying

- 400BC Chinese learned to fly a kite
- Lead to aspirations for human to fly
- Several inventions and innovations followed for centuries
- Flying is more than putting wings on a machine
- We can’t copy - we’ve to figure out what works
Software Development

- Still a nascent field
- Too many variables
- Innovation is not separate from construction
  - Separating design and coding phase is not realistic
- Capers Jones studies large software projects
  - Only 10% of projects were successful
  - We can’t afford to continue at this rate

Engineering Rigor

- In Engineering Construction is expensive, Design is relatively Cheap
- In Software Development Construction is Cheap (it’s the conversion of code into executables)
- Design (which involves modeling and coding) is expensive
- Can’t we quickly test our design (since construction is cheap)?
- Testing is the Engineering Rigor in Software Development
We’ve tried several approaches

Waterfall, Fountain, Spiral, Iterative and Incremental, Agile,...
Waterfall—pros and cons

- Simple (simplistic)
- Easy to plan
- Hard to deliver
- Assumes stages carried out to completion
- Most practiced
- High rate of failure

What’s Agility?

- What’s Agility?
  - It’s being agile
- OK, what’s Agile?
  - “marked by the ready ability to move with quick easy grace”
  - “having a quick resourceful and adaptive character”
Why Agile?

- Software Development is
  - risky
  - change is the only constant
  - we constantly have to fight entropy
  - always in a state of flux

- Conventional approach has not solved our problems

  "Walking on water and developing software from a specification are easy if both are frozen,“—Edward V. Berard.

Reliability on Estimates

From Agile and Iterative Development: A Managers Guide by Craig Larman
Change in Requirements

From Agile and Iterative Development: A Managers Guide by Craig Larman

Relevance

Actual Use of Requested Features
From Agile and Iterative Development: A Managers Guide by Craig Larman
Impact

In a waterfall lifecycle, high-risk issues such as integration and load test are tackled late.

Factors

<table>
<thead>
<tr>
<th>Success Factor</th>
<th>Weight of Influence</th>
</tr>
</thead>
<tbody>
<tr>
<td>User involvement</td>
<td>20</td>
</tr>
<tr>
<td>Executive support</td>
<td>15</td>
</tr>
<tr>
<td>Clear business objectives</td>
<td>15</td>
</tr>
<tr>
<td>Experienced project manager</td>
<td>15</td>
</tr>
<tr>
<td>Small milestones</td>
<td>10</td>
</tr>
</tbody>
</table>

From Agile and Iterative Development: A Managers Guide by Craig Larman
Duration

From Agile and Iterative Development: A Managers Guide by Craig Larman

Meeting Requirements

From "Practices of an Agile Developer" by Venkat Subramaniam and Andy Hunt
Project & Schedule

Start  Realization  Deadline  Delivery

Time

Quality  Scope

Adaptive Planning

- “No plan survives contact with the enemy” - Helmuth von Moltke

- It is more important to succeed than stick with a predefined plan

- Allow your management to dictate only two out of three - quality, time, scope

- What if they insist you give them all three?
  - They get failure instead
How to be agile?

- Agility is all about action
- How can you be evolutionary?
- You need to build what’s relevant
- You need to make change affordable
- How can you do that?

Feedback and Communication

- Actively listen and seek feedback
- Feedback comes in two forms
  - Is your code meeting and continuing to meet your (programmers’) expectations?
    - Unit and integration tests
  - Is it relevant and solving customers’ problems?
    - Frequent Demo and Exercise
Continuous, not Episodic

From "Practices of an Agile Developer"
by Venkat Subramaniam and Andy Hunt