Essence of Agility

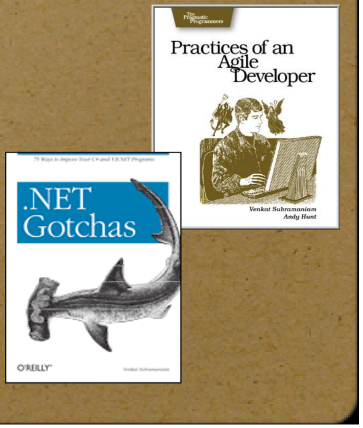
spkr.name = 'Venkat Subramaniam'

spkr.company = 'Agile Developer, Inc.'

spkr.credentials = %w{Programmer, Trainer, Author}

spkr.blog = 'agiledeveloper.com/blog'

spkr.email = 'venkats@agiledeveloper.com'



Abstract

- So what does it take to be agile, on your software projects, that is? Is it unit testing? Is it having those stand-up meetings? What does "we're on an agile project" really mean? In this presentation, we will discuss agility and look at some approaches and (.NET) tools that can help us get there.

- Along the way, we'll walk through 10 essential steps to being agile.

Agenda

- What’s Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

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?

- 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

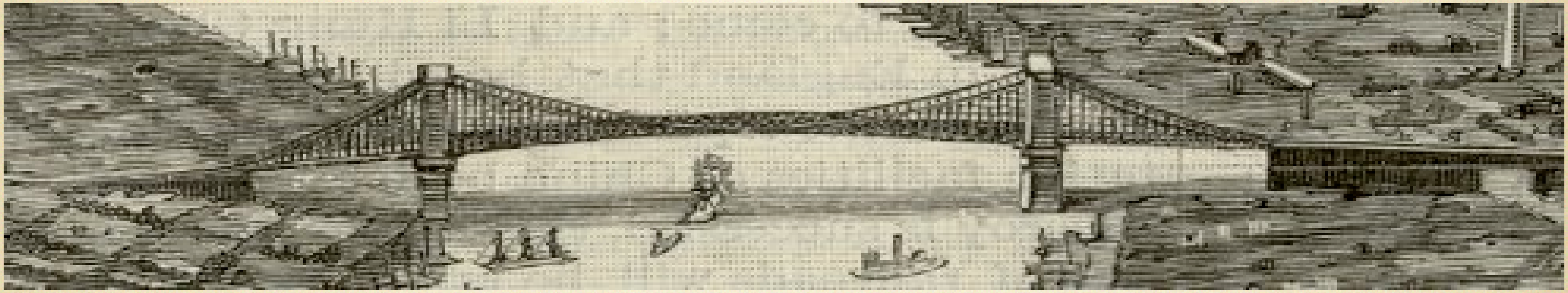
Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

Software Development

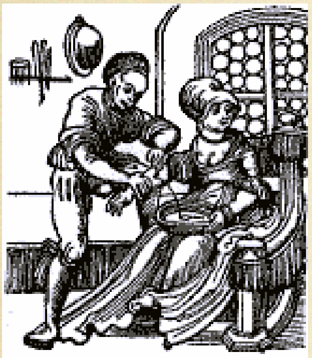
- What's software development like?
- We often get compared to other human endeavors
- Let's study some of those
  - Bridge Construction
  - Medicine
  - Flying

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

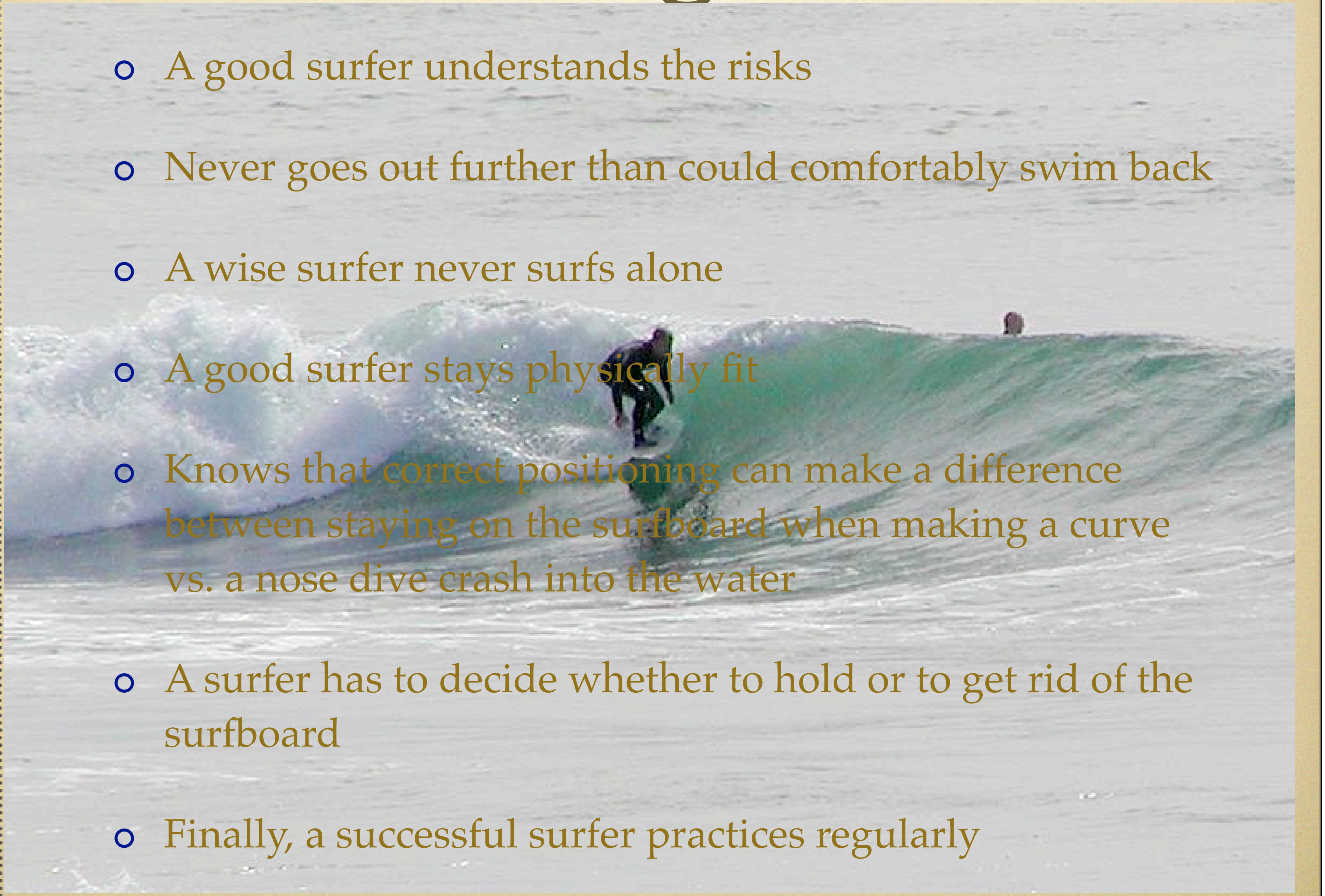
- Still a nascent field
- Too many variables
- Innovation is not separate from construction
  - Separating design and coding phase is not realistic
- Capers Jones studies large software projects
  - Only 10% of projects were successful
- We can’t afford to continue at this rate

What’s SW Dev Like?

How's that?

- What makes surfing so challenging?
  - Dynamic ever changing environment
  - Sea is unpredictable, risky, sharks in water,
  - Every wave is different.
  - Waves take unique shape and behavior based on locale
- What makes software development so challenging
  - Requirements and challenges are your waves
  - Never ceasing and ever-changing.
  - Like waves, projects take different shapes and pose different challenges
  - And sharks come in many different guises

What makes good surfer?



- A good surfer understands the risks
- Never goes out further than could comfortably swim back
- A wise surfer never surfs alone
- A good surfer stays physically fit
- Knows that correct positioning can make a difference between staying on the surfboard when making a curve vs. a nose dive crash into the water
- A surfer has to decide whether to hold or to get rid of the surfboard
- Finally, a successful surfer practices regularly

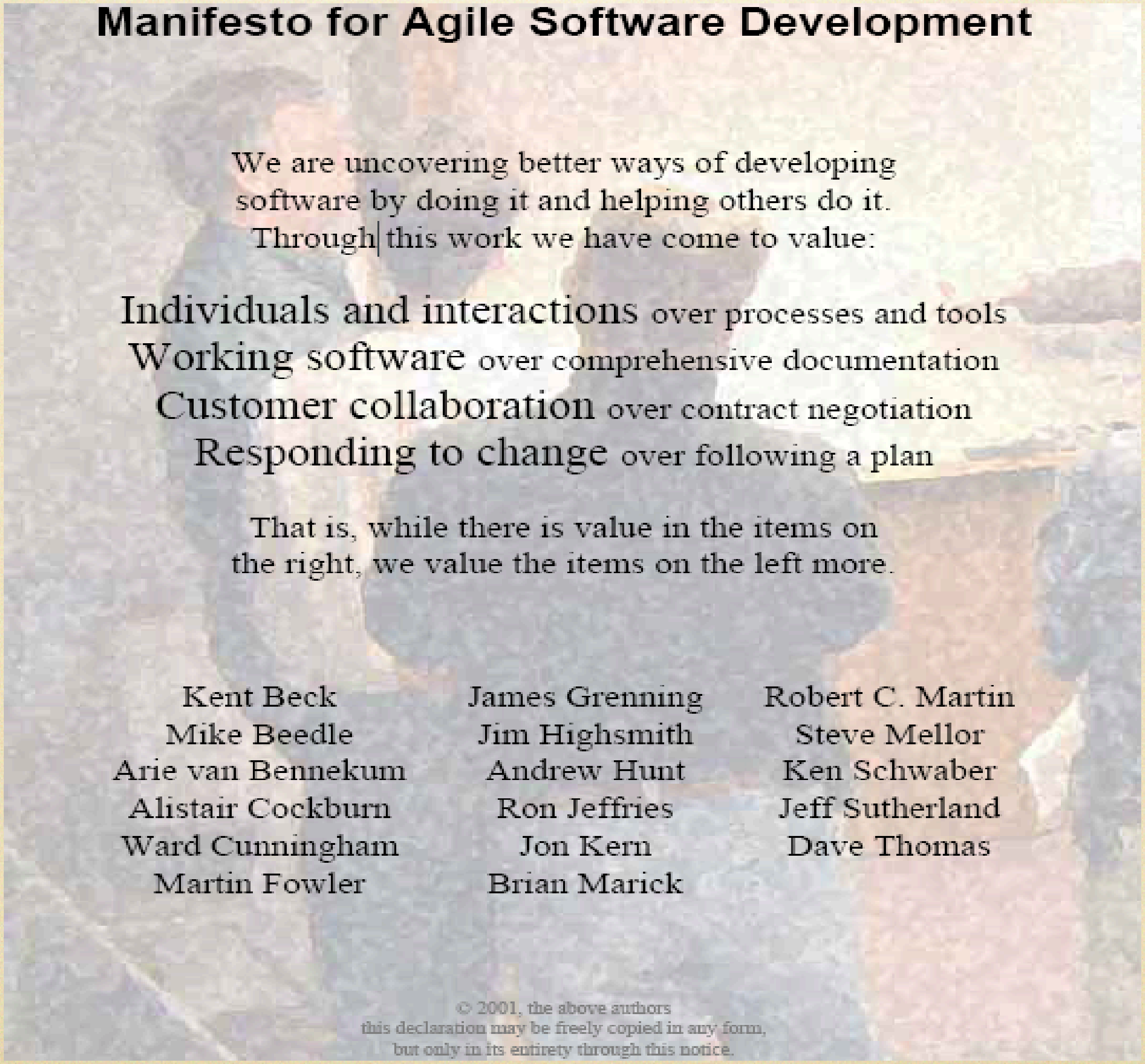
Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

Agility in Development

- There are some rigorous development practices
- Then there are approaches that have worked
- Some of us have tried those secretively
- A few have realized that there are better, lightweight, more pragmatic way
- A few of them gathered in Snowbird, Utah in 2001

Agile Manifesto



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

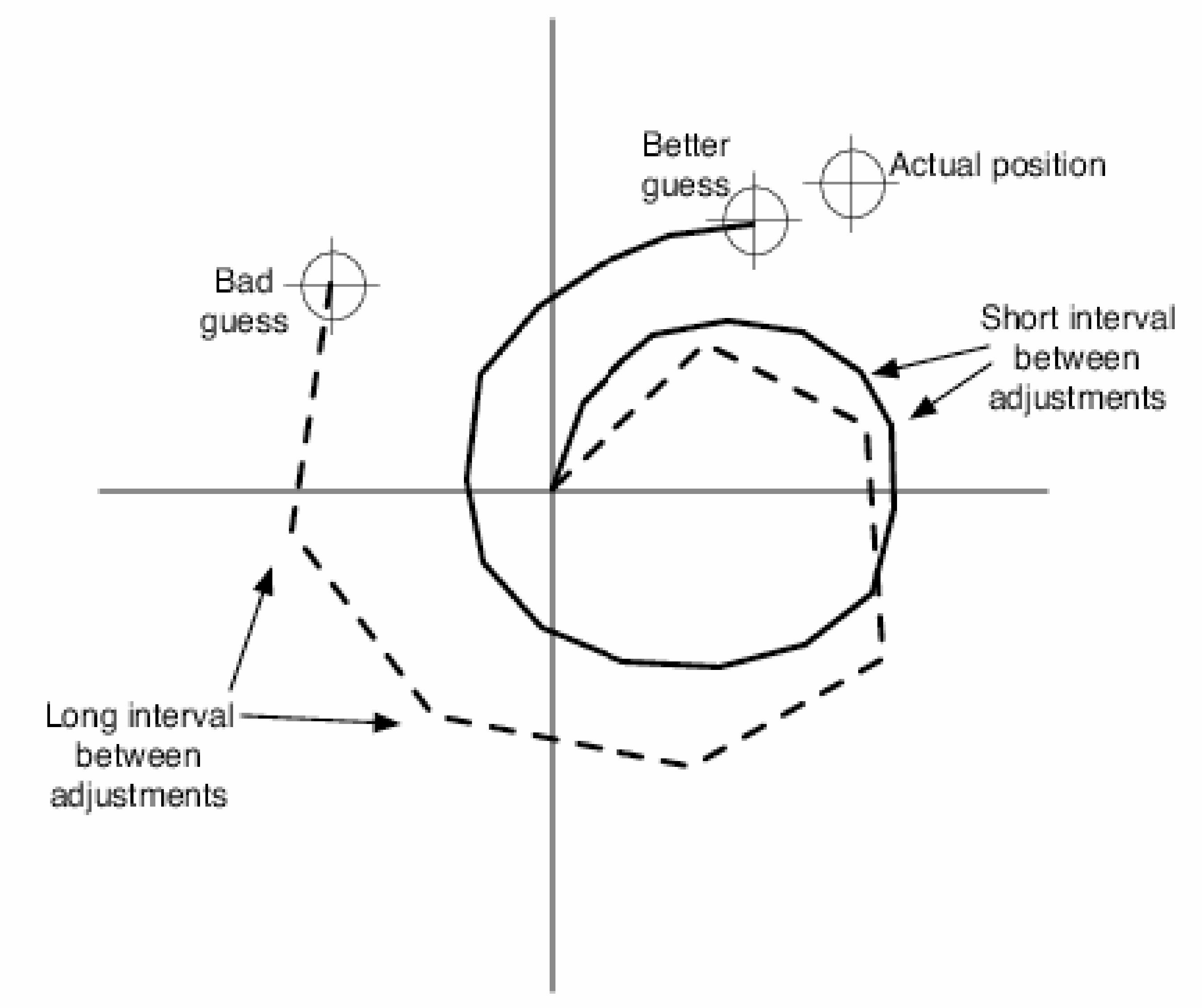
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

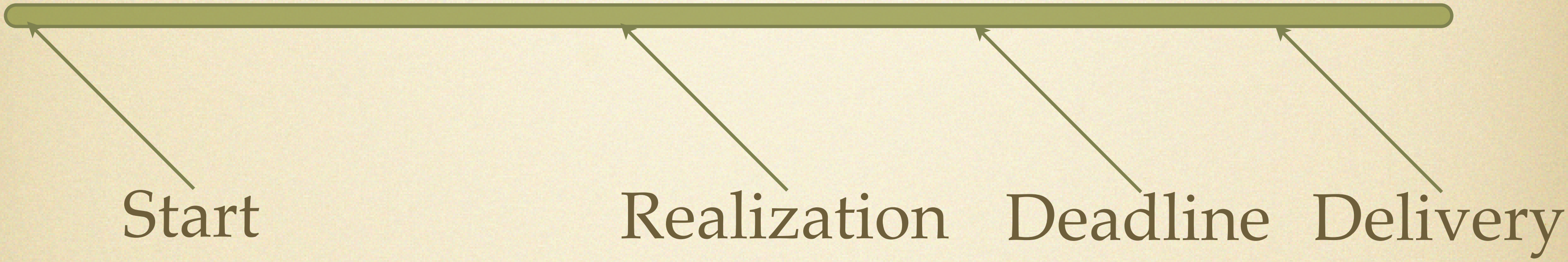
Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- ✺ Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

Meeting Requirements


Better guess
Actual position
Bad guess
Short interval between adjustments
Long interval between adjustments

Project & Schedule

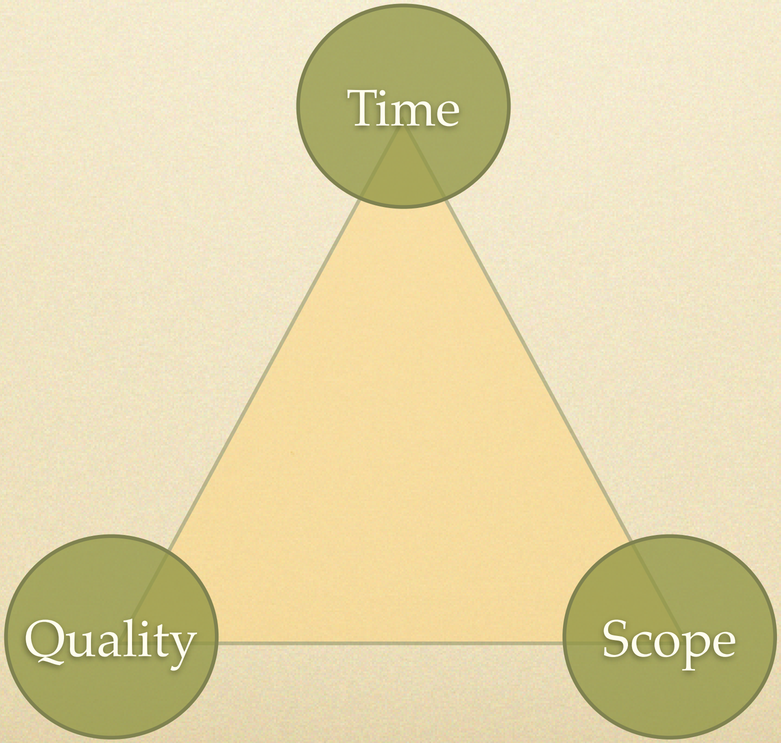


Start

Realization

Deadline

Delivery



Time

Quality

Scope

Adaptive Planning

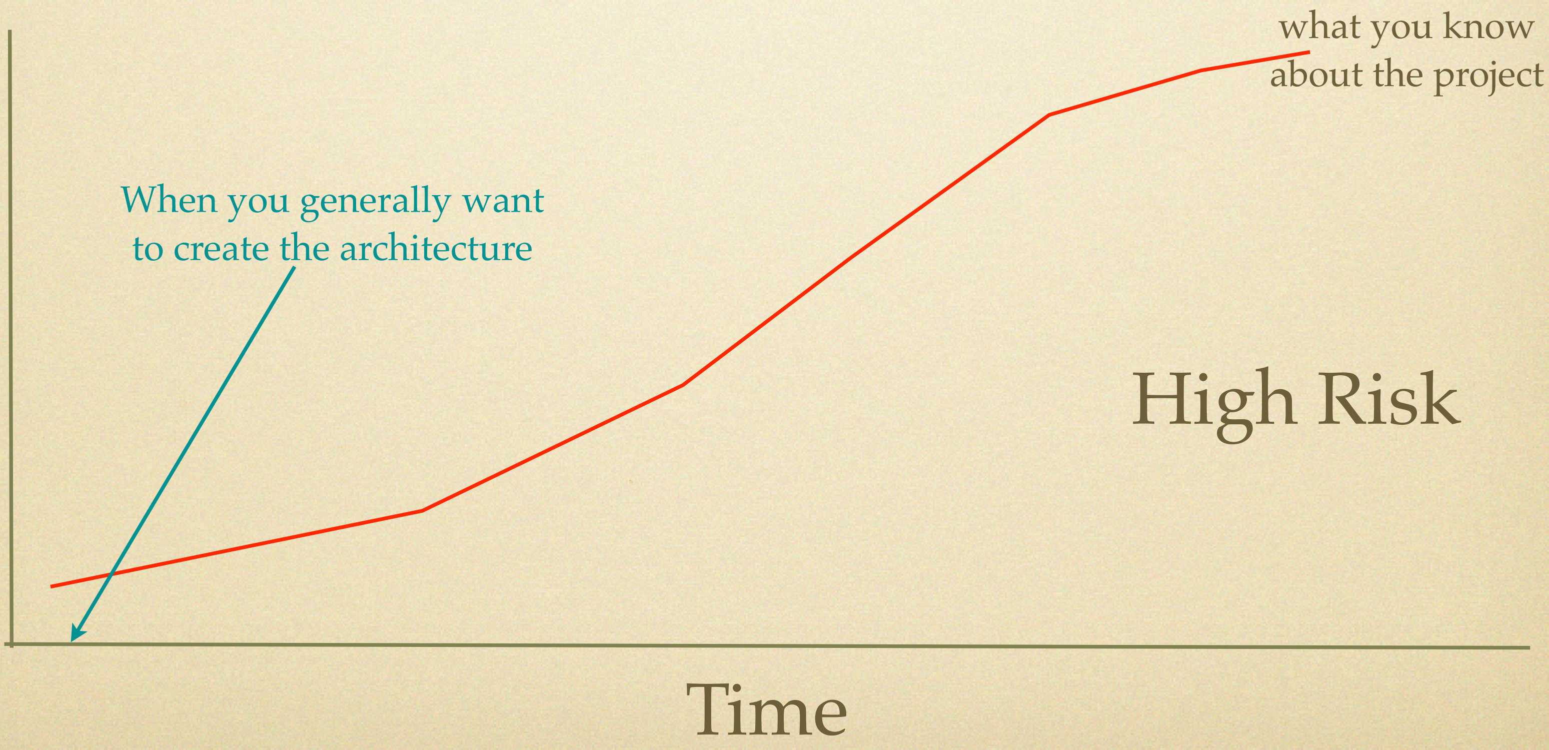
- “No plan survives contact with the enemy” - Helmuth von Moltke
- It is more important to succeed that stick with a predefined plan
- Allow your management to dictate only two out of three - quality, time, scope
- What if they insist you give them all three?
  - They get failure instead

Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

Agility and Architecture

- What's Architecture?
  - High level design of subsystem and connectivity
- You don't want to get it wrong!



what you know about the project

When you generally want to create the architecture

High Risk

Time

Allow Architecture to evolve...

Agile == No Design

Myth

- Agile Development does not discourage design
  - It discourages all up-front design
  - It encourages evolutionary design
- Design is critical
- Without it, you're seeking fragility and not agility
- You constantly evaluate and evolve design
  - Following good design principles and refactoring

Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

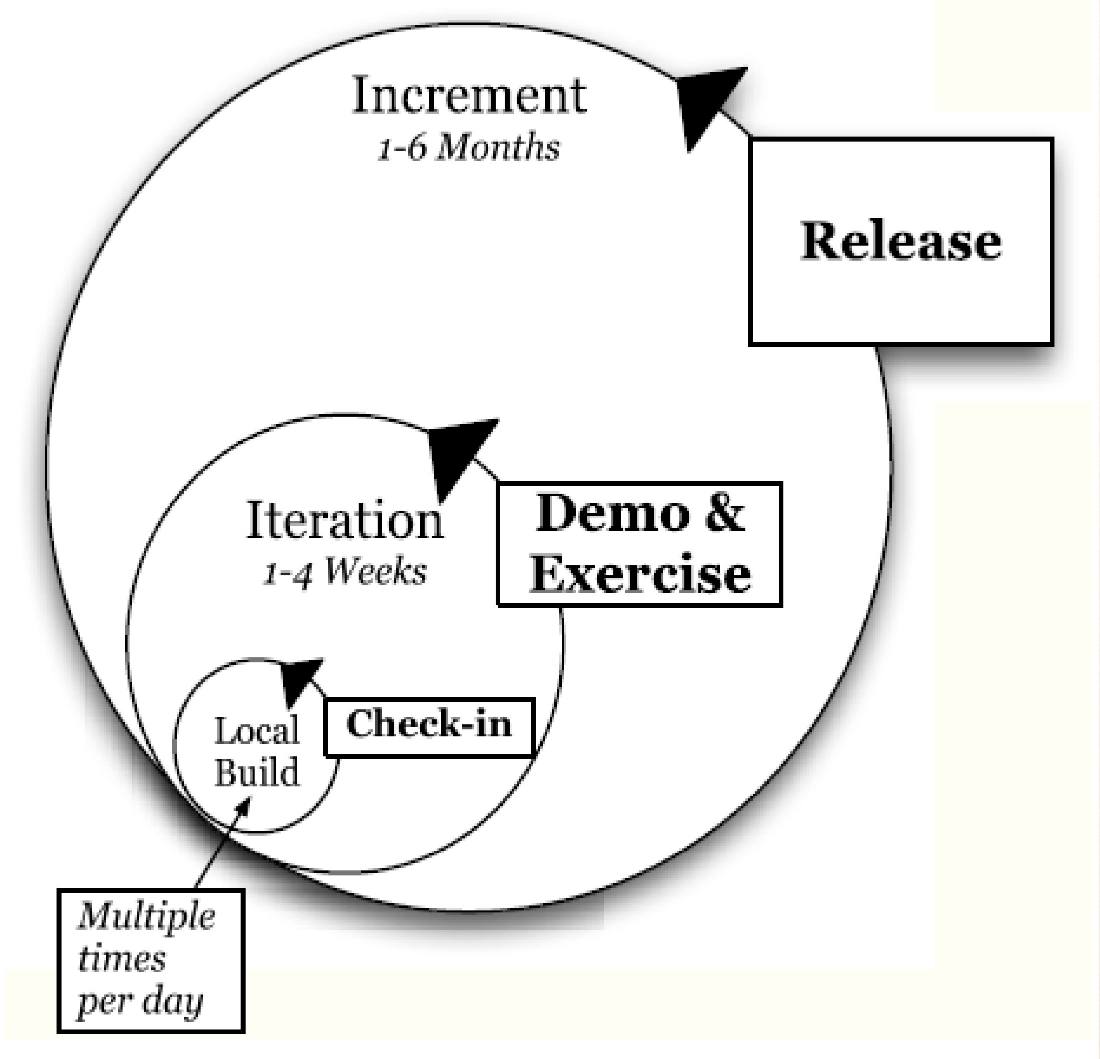
How to be agile?

- 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



Care About Code

Broken Window Theory

http://en.wikipedia.org/wiki/Fixing_Broken_Windows

- Do not let anyone trash your application
- Make sure application is always releasable
- Get continuous feedback
  - If something falls apart, know and act on it right away

Tools to help us

- NUnit
- Mock Objects

  (EasyMock, NMock,...)
- NCover (Code coverage)
- Nester

- NDepend
- Fit/FitNess
- Selenium
- Cruise Control .NET
- XPlanner, ...

Visual Studio Team System has a number of these capabilities built in

Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

Agile Team

- Attitude makes a big difference
- Smaller teams better than larger teams
- Where possible, face to face conversation
- Passionate developers with great attitude miles apart is better than neighbors who hate each other
- Team owns the code - collective ownership
- Self directed, highly competent teams
- Respect and Responsibility

Agenda

- What's Agility and Why?
- State of development
- Agile Movement
- Adaptive Planning
- Architecture, Design, and Evolution
- Coping with change
- Agile Team
- Essence of Agility

Essence of Agility

1. Build Relevant Working Software
2. Seek continuous feedback from customers
3. Keep application healthy and releasable at all times
4. Be continuous, not episodic
5. Evolve your architecture
6. Make evolution affordable
7. Do not design in isolation
8. Practice Collective Ownership
9. Ensure that Customers Exercise your application regularly
10. Check your Ego at the door - work to solve problems

Thank You!

http://www.agiledeveloper.com/download.aspx