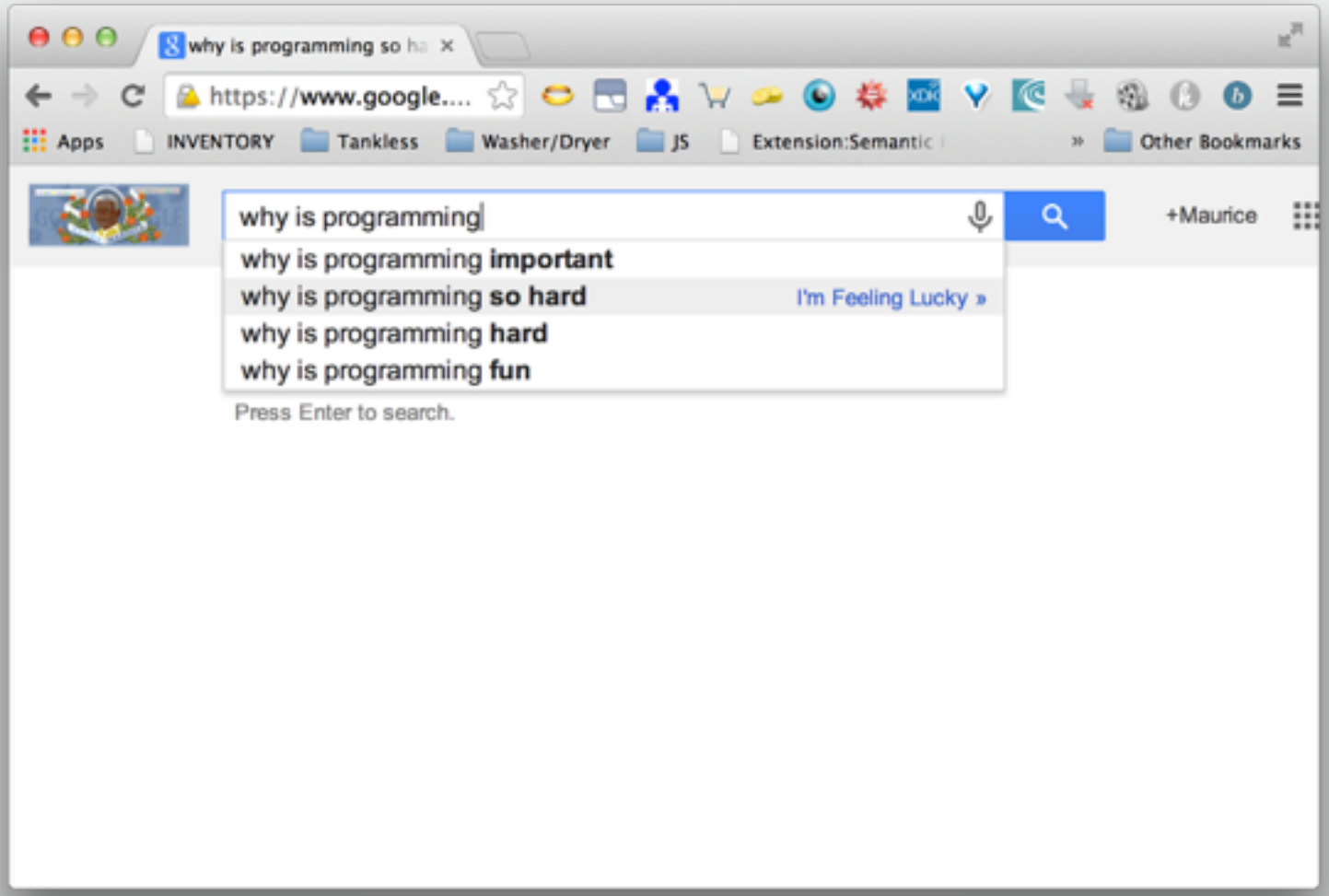
Agile Software Development

Author: Maurice Rabb

Phase 2: Week 1 - Day 5

Guess what?


why is programming so ha
https://www.google....
Apps
INVENTORY
Tankless
Washer/Dryer
JS
Extension:Semantic
Other Bookmarks
why is programming
why is programming important
why is programming so hard
why is programming hard
why is programming fun
I'm Feeling Lucky »
Press Enter to search.
+Maurice

Why is programming so hard?

http://rebloggy.com/post/lol-wat/36994389573

Software development is hard

- Programming languages are brittle
- Software is complex
- Requirements change
- Technologies change
- Estimating is hard
- Working with people is ~~nothing but unicorns and rainbows~~ challenging

Software Methodology

- Be smart
- Be deliberate
- Be disciplined

Methodology explosion

- 70's
  - **Waterfall**
  - Top-down (Niklaus Wirth)
  - The Mythical Man-Month (Fred Brooks)
    - "Adding manpower to a late software project makes it later."
- 80's
  - No Silver Bullet paper (Fred Brooks)
  - Object-oriented analysis
- 90's
  - Dynamic systems development method (DSDM)
  - Rational Unified Process (RUP) + UML
  - Scrum
  - eXtreme Programming
  - Test Driven Development (TDD)
- 00's
  - **Agile**
  - Behavior Driven Development (BDD)
  - Domain Driven Development (DDD)

C & UNIX '72

SQL '74

Smalltalk '80

SUnit '89

PostgreSQL '95

JavaScript '95

Ruby '95

Active Record '03

jQuery '06

RSpec '07

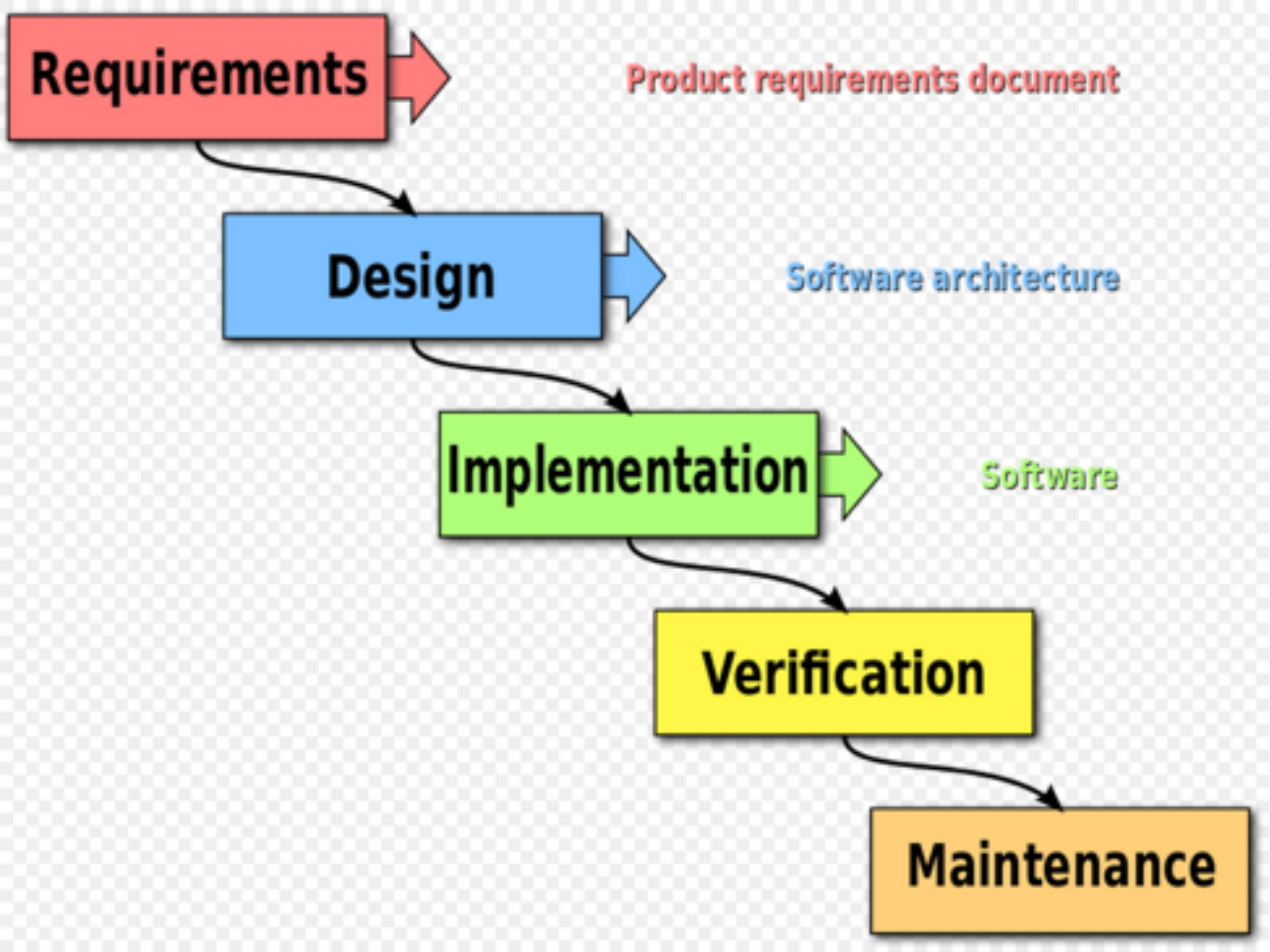
ECMAScript 5 '09

Waterfall

http://www.conservationinstitute.org/10-biggest-largest-waterfalls-in-the-world

Nigara Waterfall (Rodnet Campbell/Flickr)

Waterfall



http://inthralld.com/2012/12/worlds-largest-kaleidoscope-made-of-swarovski-crystal/

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

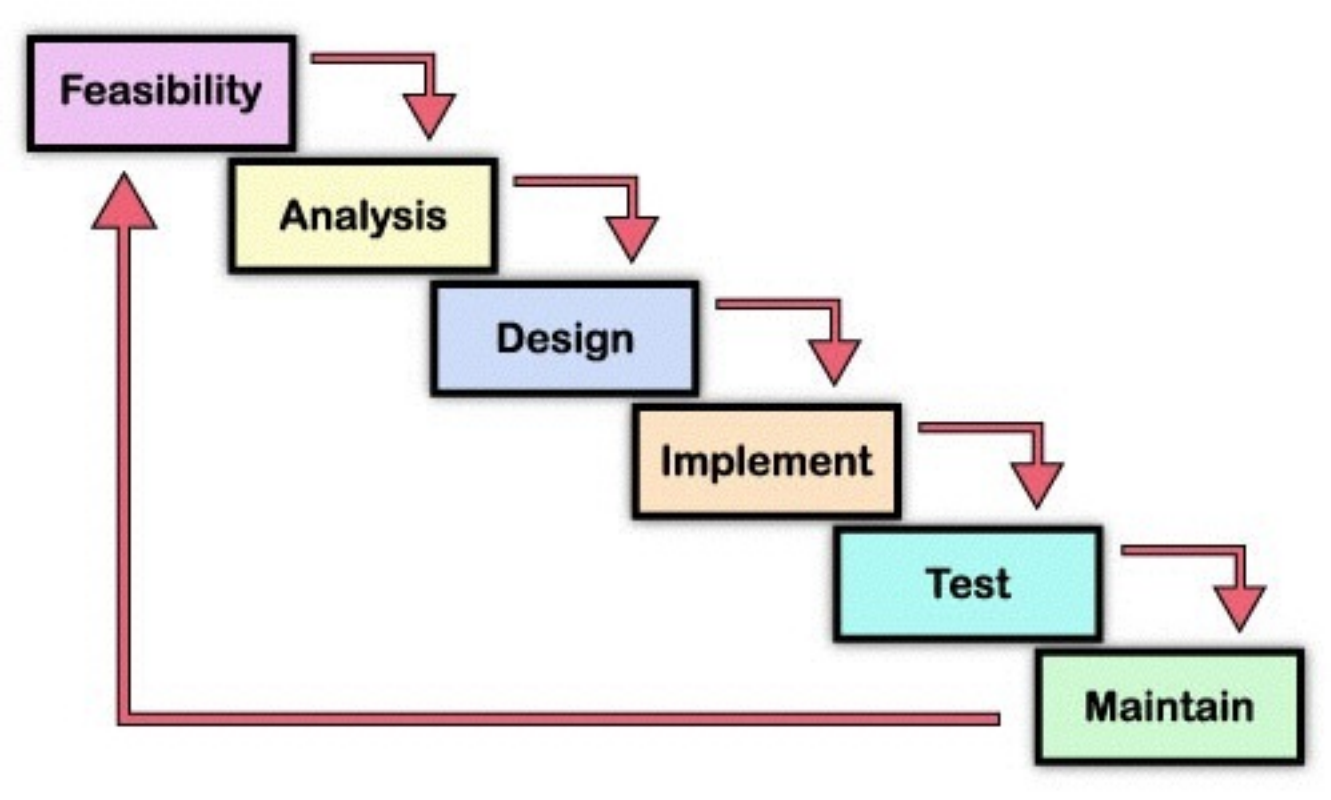
Waterfall trickles down

http://inthralld.com/2012/12/worlds-largest-kaleidoscope-made-of-swarovski-crystal/

What's missing is *feedback*

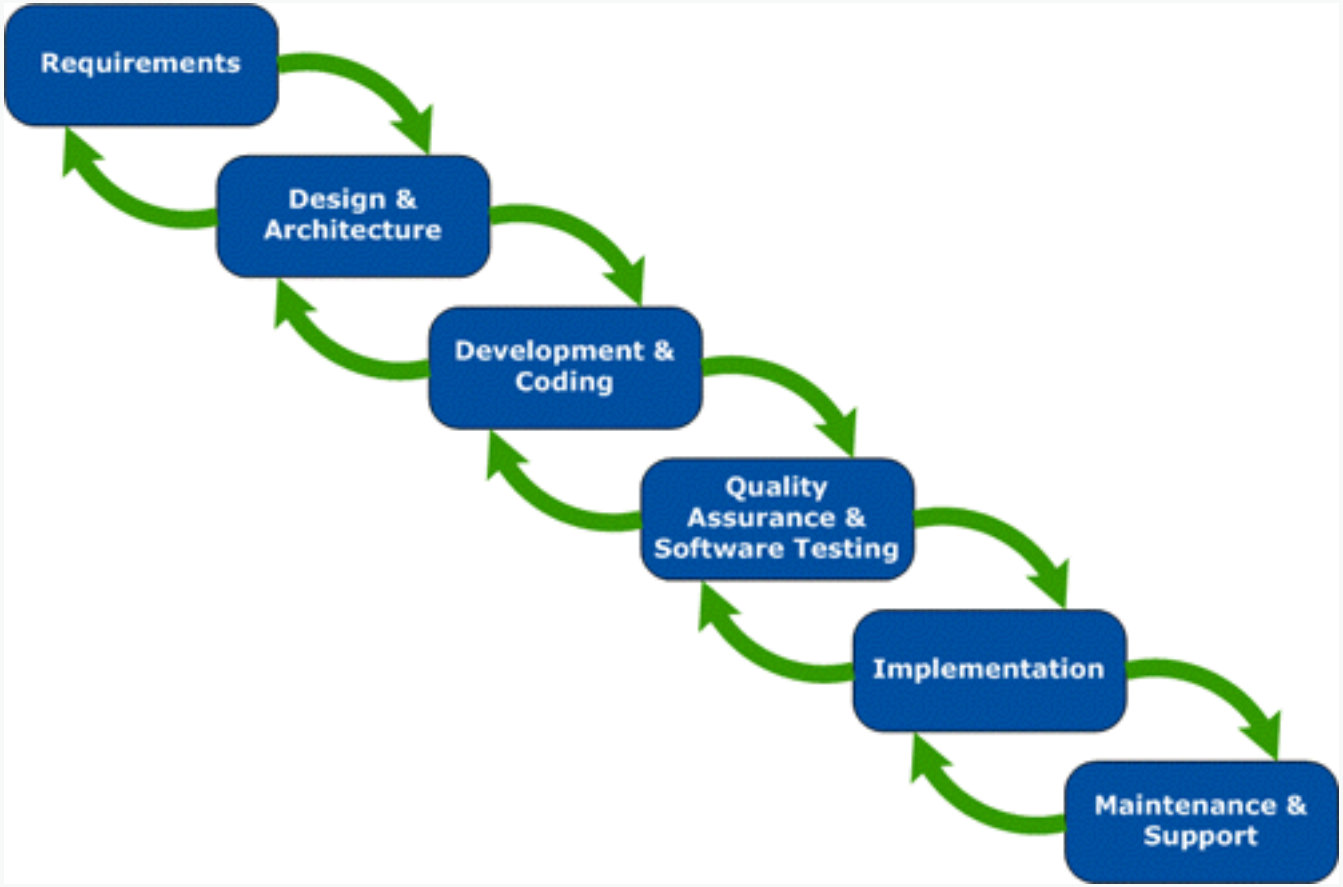
http://satriasputra.blogspot.com/2010/04/kolam-renang-setan-hanya-ada-di-afrika.html

Waterfall with *feedback*



http://www.websoftwareqa.com/2011/01/applying-qa-to-waterfall-development-wait-what/

But wait! Waterfall, now with *incremental* feedback



http://www.managedmayhem.com/wp-content/uploads/2009/05/developmentprocess.gif

Mighty Waterfall!

http://www.freetraveltalk.com/america/south-america/iguazu-falls-the-worlds-widest-waterfall.html

Mystical Waterfall!!

http://www.mymodernmet.com/profiles/blogs/breathtaking-rainbows-over-victoria-falls

Wait, wait a moment…

http://www.worldrecordacademy.com/stunts/highest_waterfall_dive_world_record_set_by_Di_Huanran_101481.htm

That’s a whole lot of commitment!

http://www.incrediblediary.com/5-worlds-most-beautiful-and-amazing-waterfalls.html

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

Uh. Oh…

Aw hell to the naw!

Waterfall sucks. Pure misery imnsho

Waterfall's shortcomings

- Projects failed to meet deadlines
- Projects were over budget
- Projects failed to meet changing requirements
- Projects failed to meet customer expectations
- Projects failed to be delivered *at all*

Heavy weight methodologies

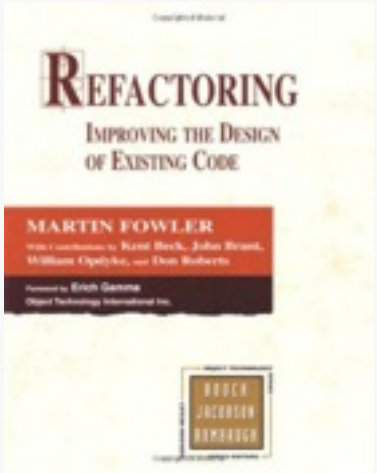
http://fate.netgame.com/community/forum/view/idxPost/32574/idxForum/8/page/2

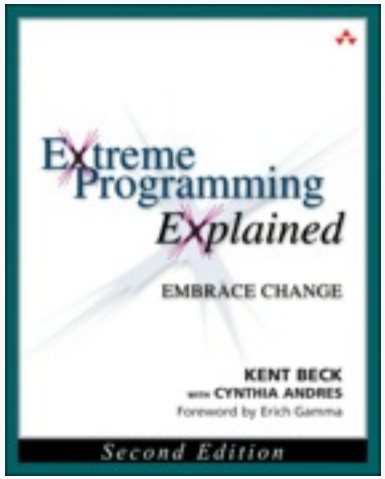
On to a more enlightened and exciting time

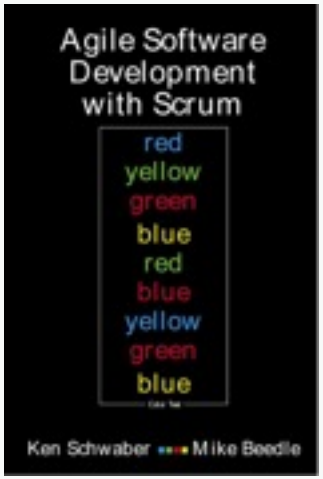
90’s - The rise of modern software engineering

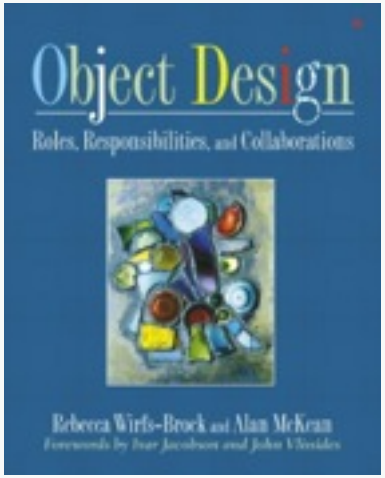
WIKI
WIKI
WEB

UNIFIED
MODELING
LANGUAGE™

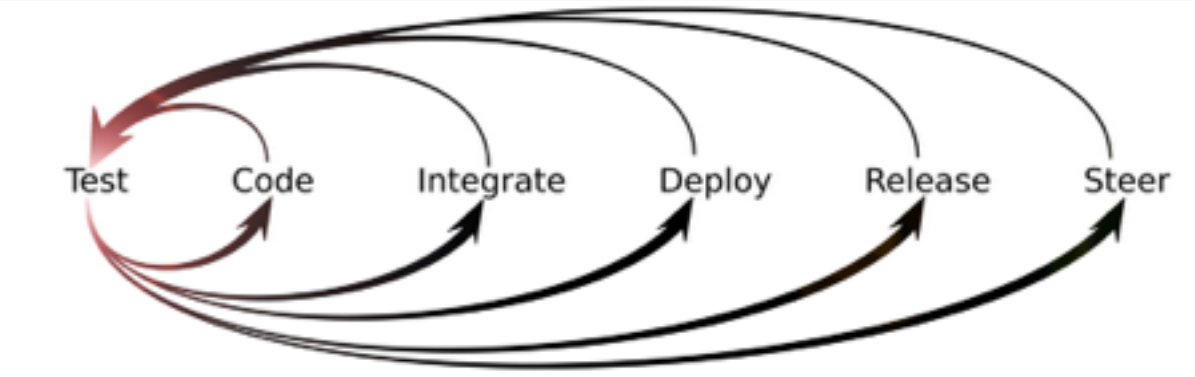




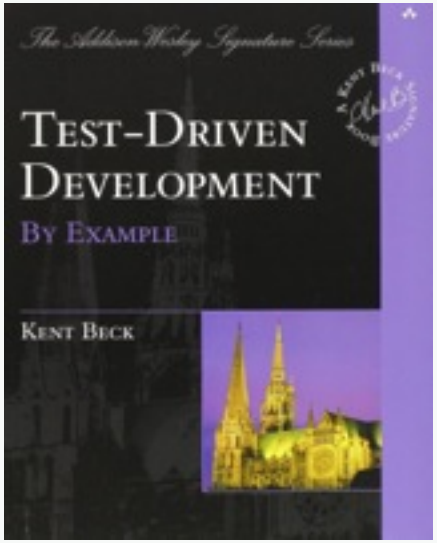


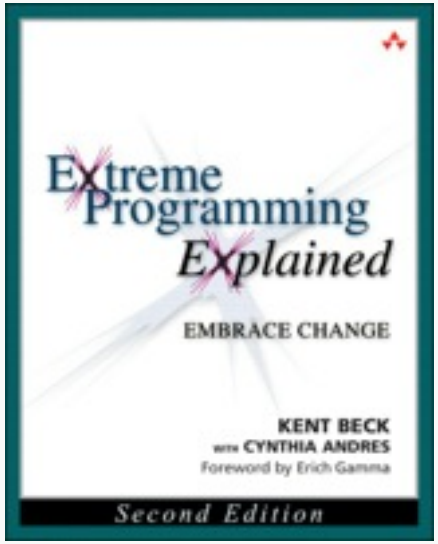


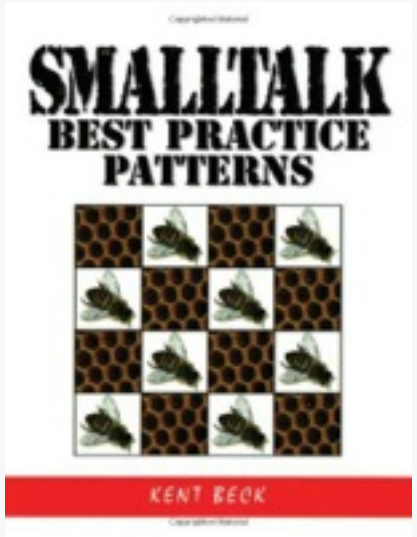
One word: @KentBeck



1) Make it work
2) Make it right
3) Make it fast.



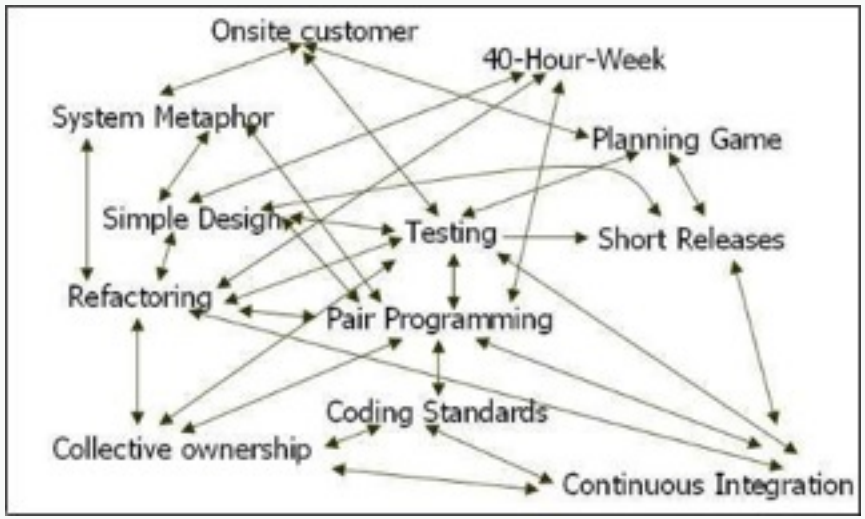




eXtreme Programming

- Values
  - communication
  - simplicity
  - feedback
  - courage

Practices



XP - if it's good, do more

- customer feedback ➙ on-site customer
- code reviews ➙ pair programming
- testing ➙ test first
- simplicity ➙ KISS, DRY, YAGNI, Do the simplest solution
- integration ➙ continuous integration
- short release times ➙ timeboxing and shipping weekly
- refactoring ➙ refactor all the time
- small methods ➙ methods *LOC* limits
- coding standards ➙ consistent code
- shared ownership ➙ indistinguishable code authorship
- documentation issues ➙ self-documenting *literate* programming

Push our values to the extreme

The Joel Test

Joel Spolsky: 12 Steps to Better Code 2000-08-09

Do you use source control?
Can you make a build in one step?
Do you make daily builds?
Do you have a bug database?
Do you fix bugs before writing new code?
Do you have an up-to-date schedule?
Do you have a spec?
Do programmers have quiet working conditions?
Do you use the best tools money can buy?
Do you have testers?
Do new candidates write code during their interview?
Do you do hallway usability testing?

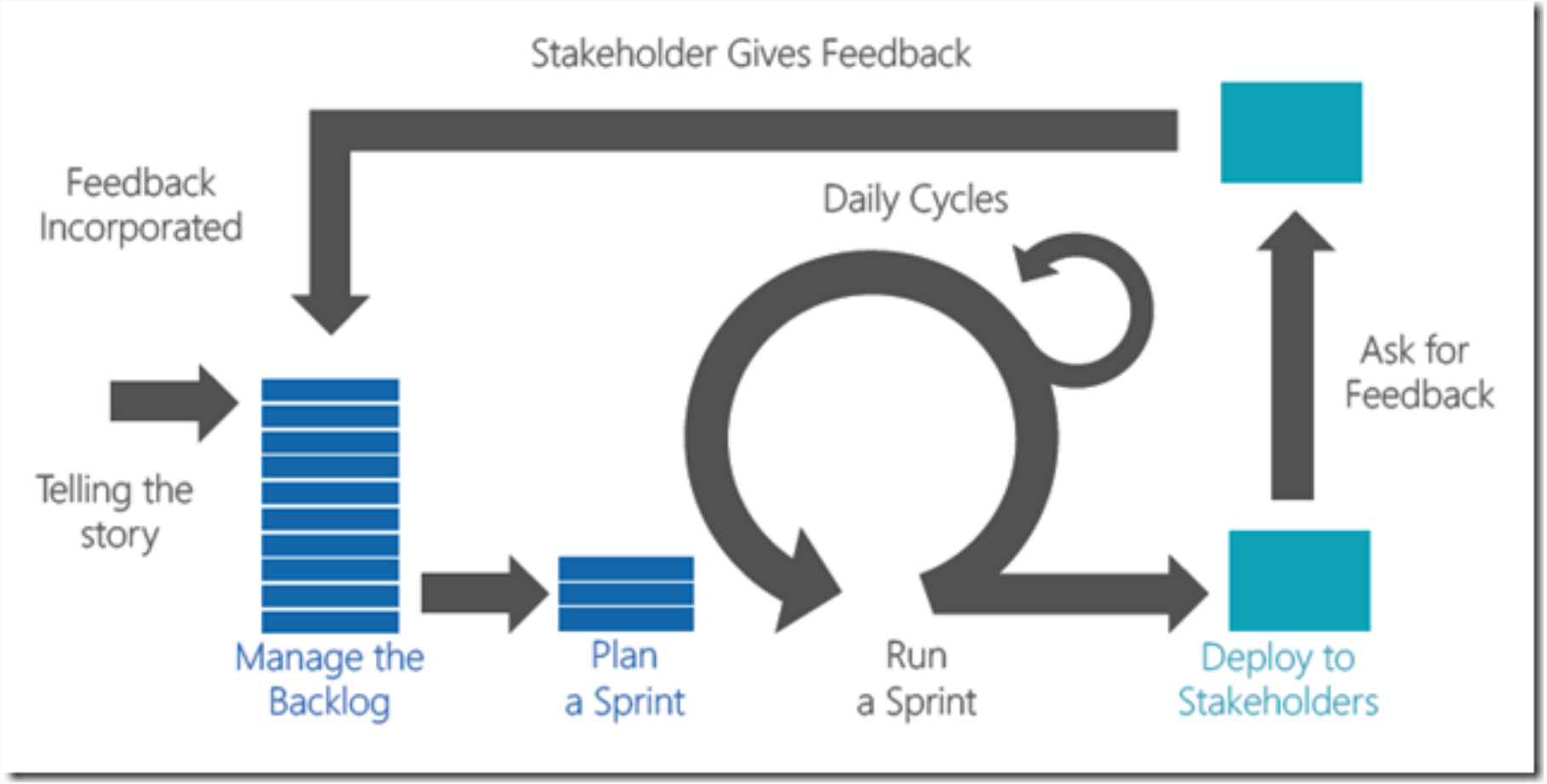
12 is perfect, 11 is tolerable, 10 or less equals serious problems

Also check out the best of *Joel on Software*

Big business is risk averse therefore XP = risky

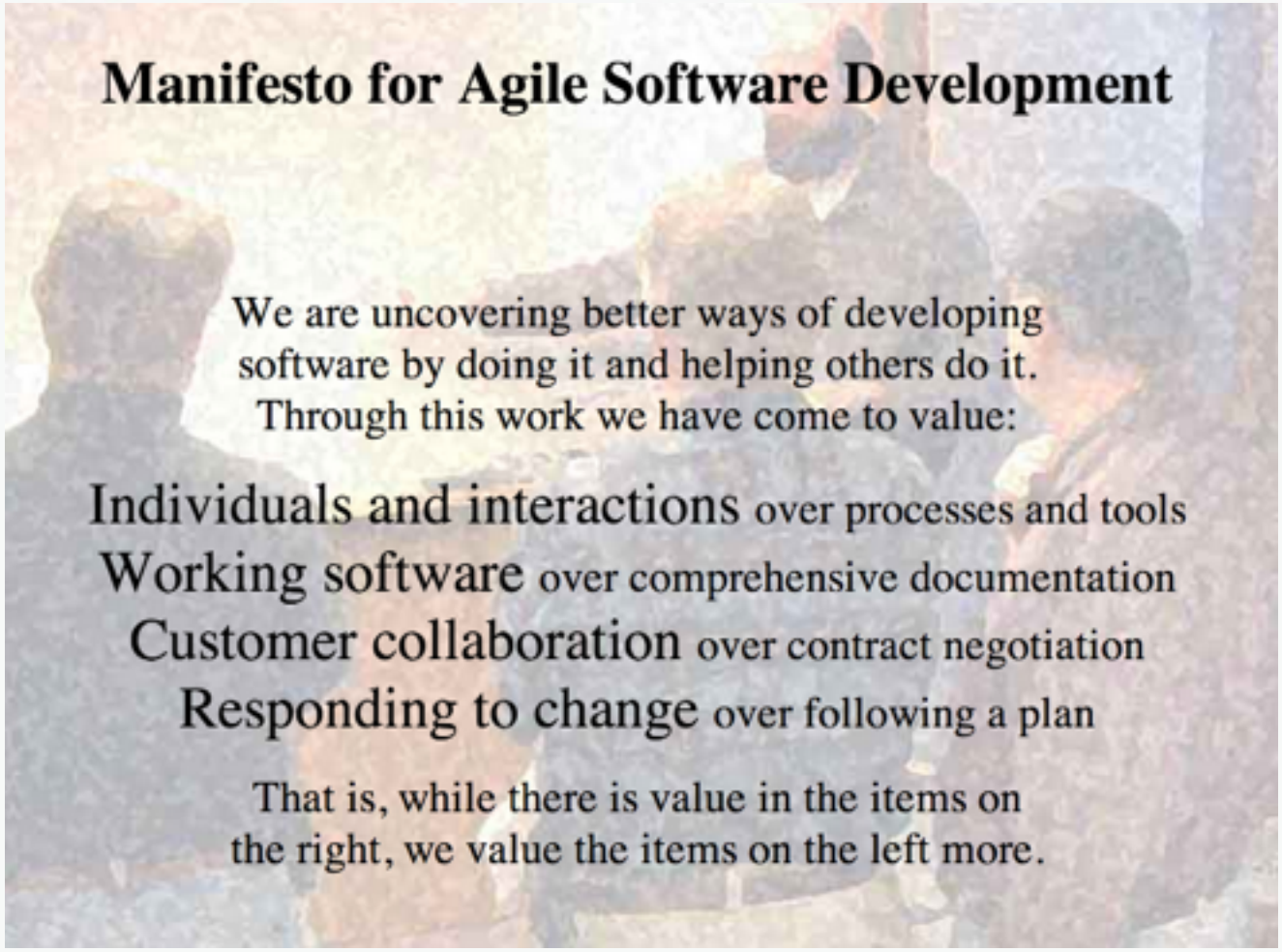
XP requires deep commitment to the process by all stakeholders

Scrum



http://kishore1021.wordpress.com/2012/08/22/creating-a-scrum-team-project-in-visual-studio-2012-using-visual-studio-scrum-2-0-process/

Agile Manifesto



Agile take aways

- adaptive vs predictive
- iterative vs waterfall
- code vs documentation
- transparent vs opaque

Employing Agile Methods

- Test Driven Development (TDD)
- Scrum boards
  - Whiteboards, notecard, stickies,
  - Trello
  - Github issues and milestones
- Planning game

The planning game

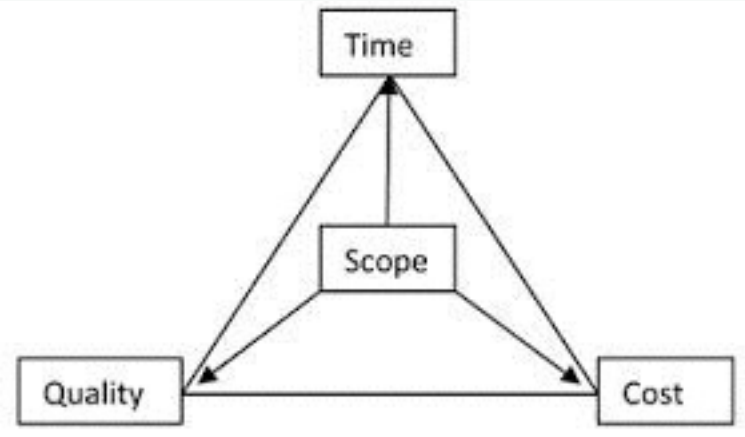
- **create user stories**
- assign difficulty
- **prioritize features**
- estimate features
- re-prioritize features
- **assign work**
- **implement code**
- evaluate results
- recalibrate *velocity*

Process

- create user stories
- create wireframes
- do standups

The art of software project management

- cost
- time
- quality
- *scope*



Now get to work!