AIRA AMERICAN IMMUNIZATION REGISTRY ASSOCIATION

Immunization Information Systems for a New Era

# AGILE VS. WATERFALL

AIRA Discovery Session
October 24, 2016
4pm Eastern

# OVERVIEW

High level overview of the differences between agile and waterfall methodology

Perspectives from the field – Informal lessons learned from IIS Vendors, Central IT, others

Questions, Comments and Discussion

# TODAY'S SPEAKER:

Nathan Bunker, AIRA Senior Technical Manager

# Agile Development Concepts

AIRA Discovery Session

Nathan Bunker

AIRA Senior Technical Project Manager

October 24, 2016

AIRA AMERICAN IMMUNIZATION REGISTRY ASSOCIATION

Immunization Information Systems for a New Era

# Government and Software

- Information management is:
  - Key government capability
  - Critical for public health
  - Our projects can not work without it
- Modern computer systems:
  - Make management of large data sets possible
  - Allow better government functions than was previously possible
- Software development is:
  - Central to the success of our projects

# Government and Software

- Expectations are high
  - We expect government to operate like Google, YouTube or Facebook
  - Government wants to install new and better software
- Barriers exist
  - Government processes are not designed to support software systems
  - Investments are probably too low or not well enough focused
  - Public health leadership is still learning how use software
  - Entrenched systems and people need time to change and come up to speed

# Agenda

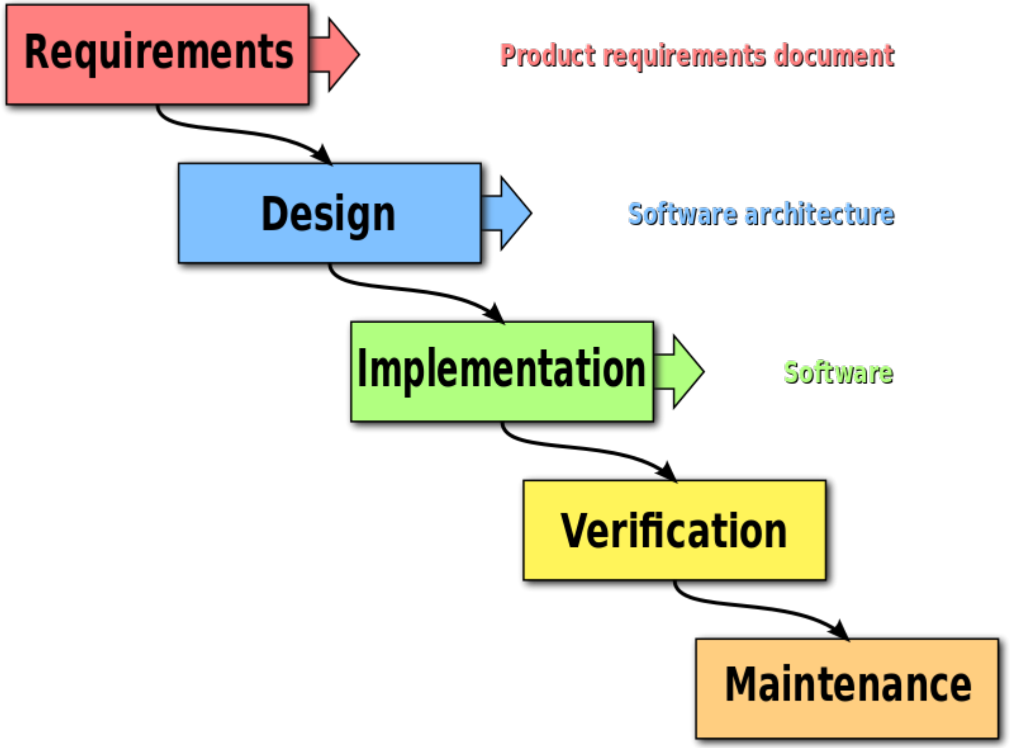
- Waterfall Method of Software Development
- Agile Philosophy
- Impact of Waterfall and Agile in Government IT

# Waterfall Model

Traditional government process for software development

# Waterfall Model

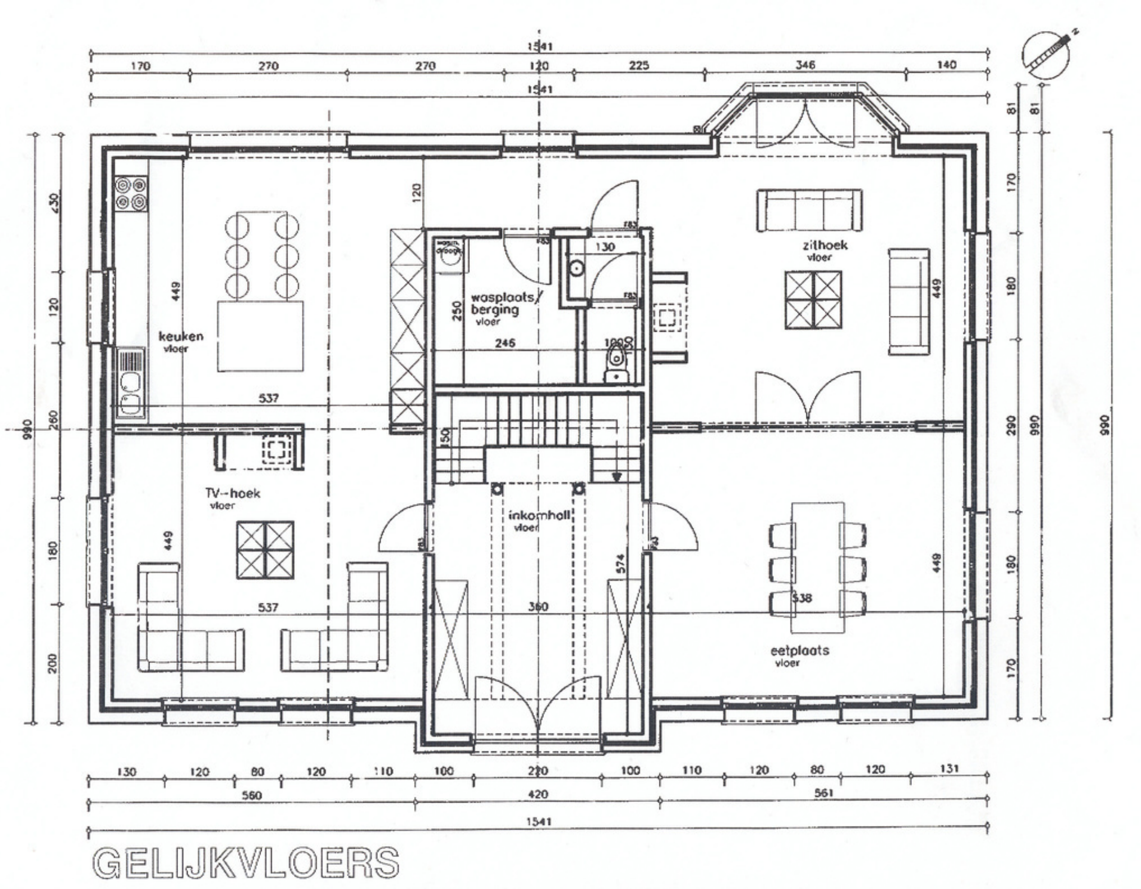
- Discrete phases
  - Each phase must be completed before another starts
- Why it works well:
  - Good initial design can reduce costs later
  - Requires detailed set of procedures and controls
  - Requires good documentation
  - Solid milestones



By Peter Kemp / Paul Smith - Adapted from Paul Smith's work at wikipedia, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=10633070

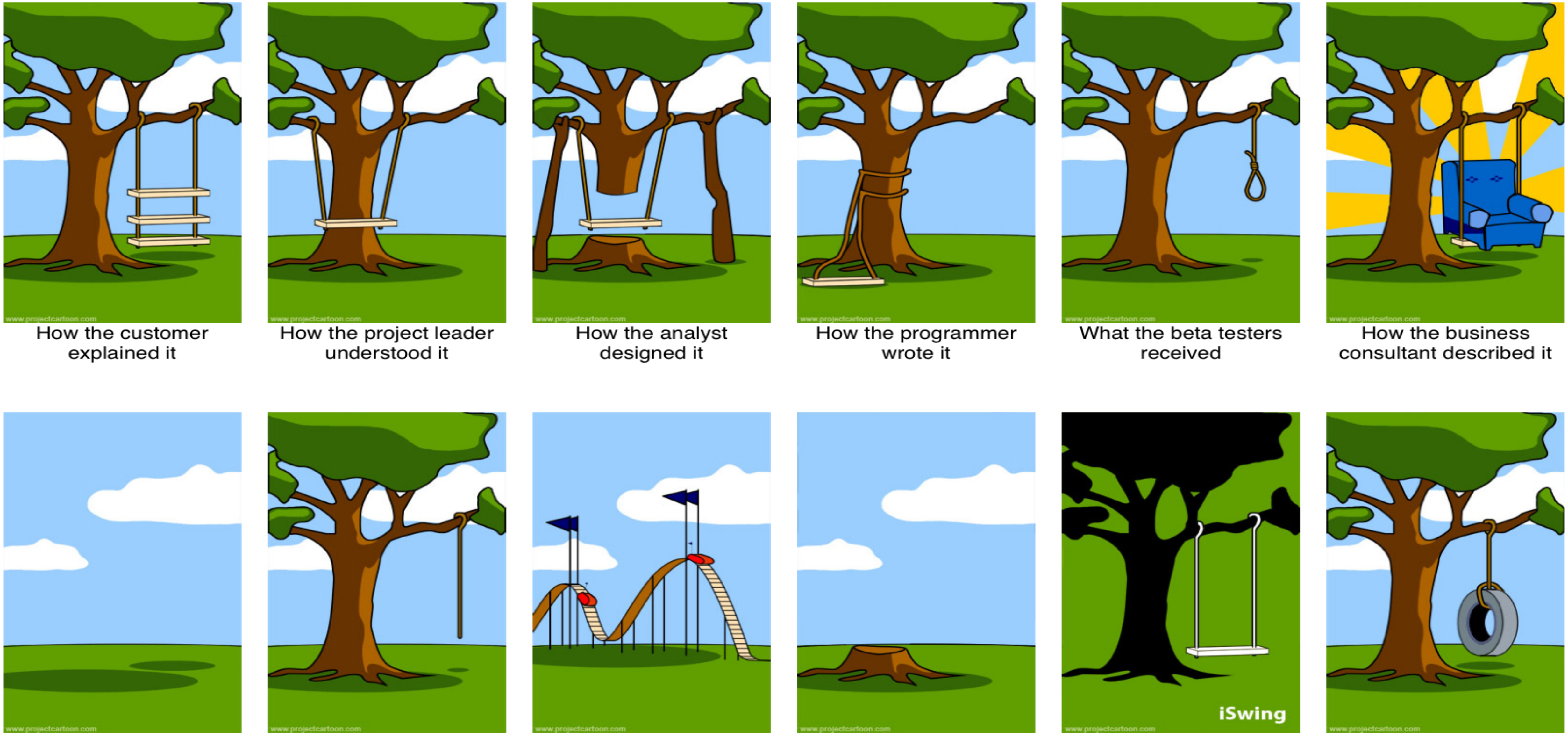
# Waterfall Model

- Well suited for projects where:
  - Requirements and scope are fixed
  - Product itself is firm and stable
  - Technology is clearly understood
- Same model used in construction
  - Requirements, Design, Implementation, Verification, Maintenance



# How Projects Really Work (version 1.5)

Create your own cartoon at www.projectcartoon.com



How the customer explained it

How the project leader understood it

How the analyst designed it

How the programmer wrote it

What the beta testers received

How the business consultant described it

How the project was documented

What operations installed

How the customer was billed

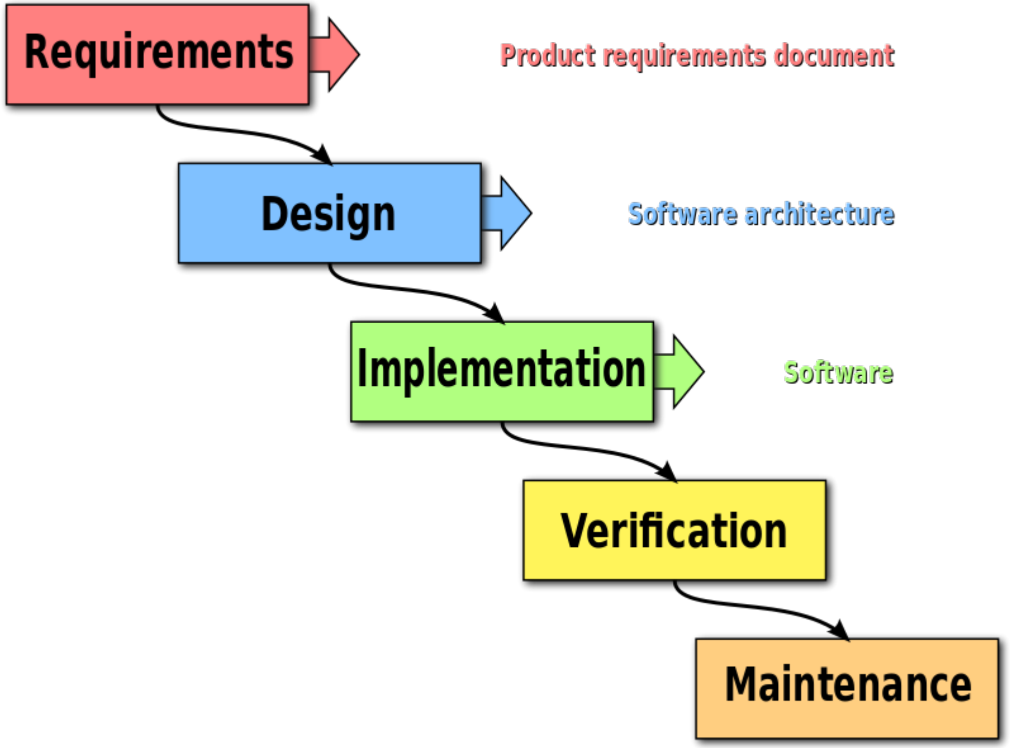
How it was supported

What marketing advertised

What the customer really needed

# Waterfall Model - Drawbacks

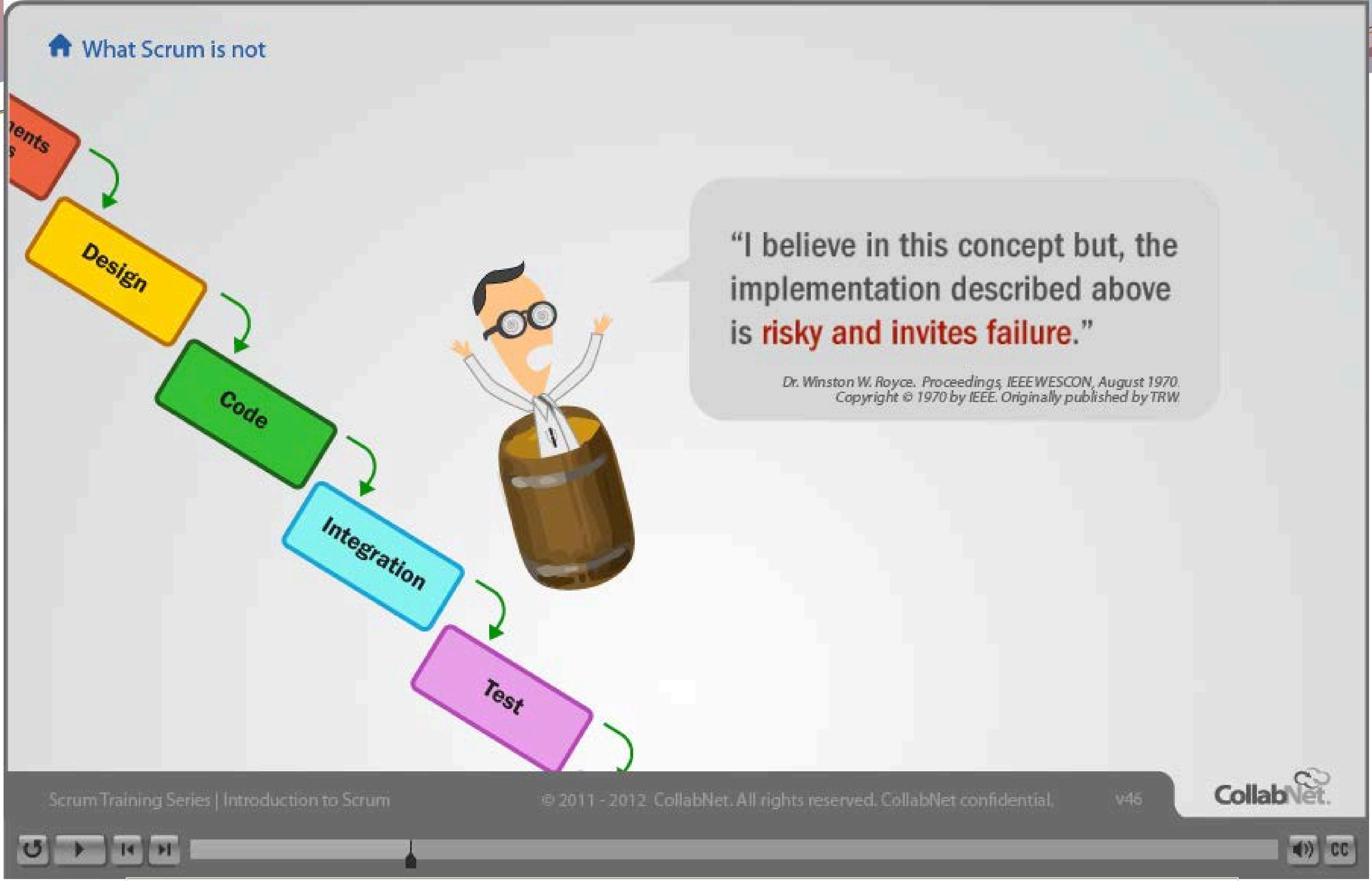
- Requirements must be fixed from the beginning
- Technical changes not expected
- Long delay before mistakes found and fixed
- Long development time before stakeholders can see impact
- No assurance final solution will be what was really needed until the project is finished



By Peter Kemp / Paul Smith - Adapted from Paul Smith's work at wikipedia, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=10633070

# Agile Software Development

How rapid feedback improves software projects



What Scrum is not

“I believe in this concept but, the implementation described above is risky and invites failure.”

Dr. Winston W. Royce. Proceedings, IEEE WESCON, August 1970.
Copyright © 1970 by IEEE. Originally published by TRW.

Scrum Training Series | Introduction to Scrum

© 2011 - 2012 CollabNet. All rights reserved. CollabNet confidential.

v46

CollabNet.

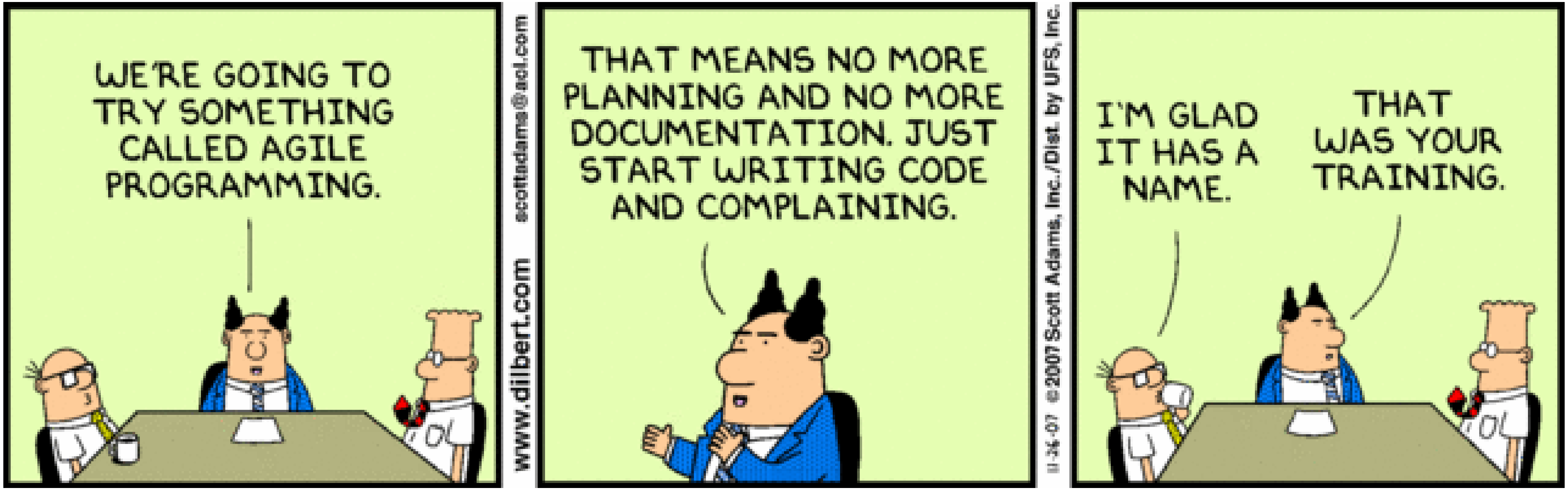
See: http://scrumtrainingseries.com/Intro_to_Scrum/Intro_to_Scrum.htm

# 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:

- From http://agilemanifesto.org/
  - **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

# How do you implement Agile?

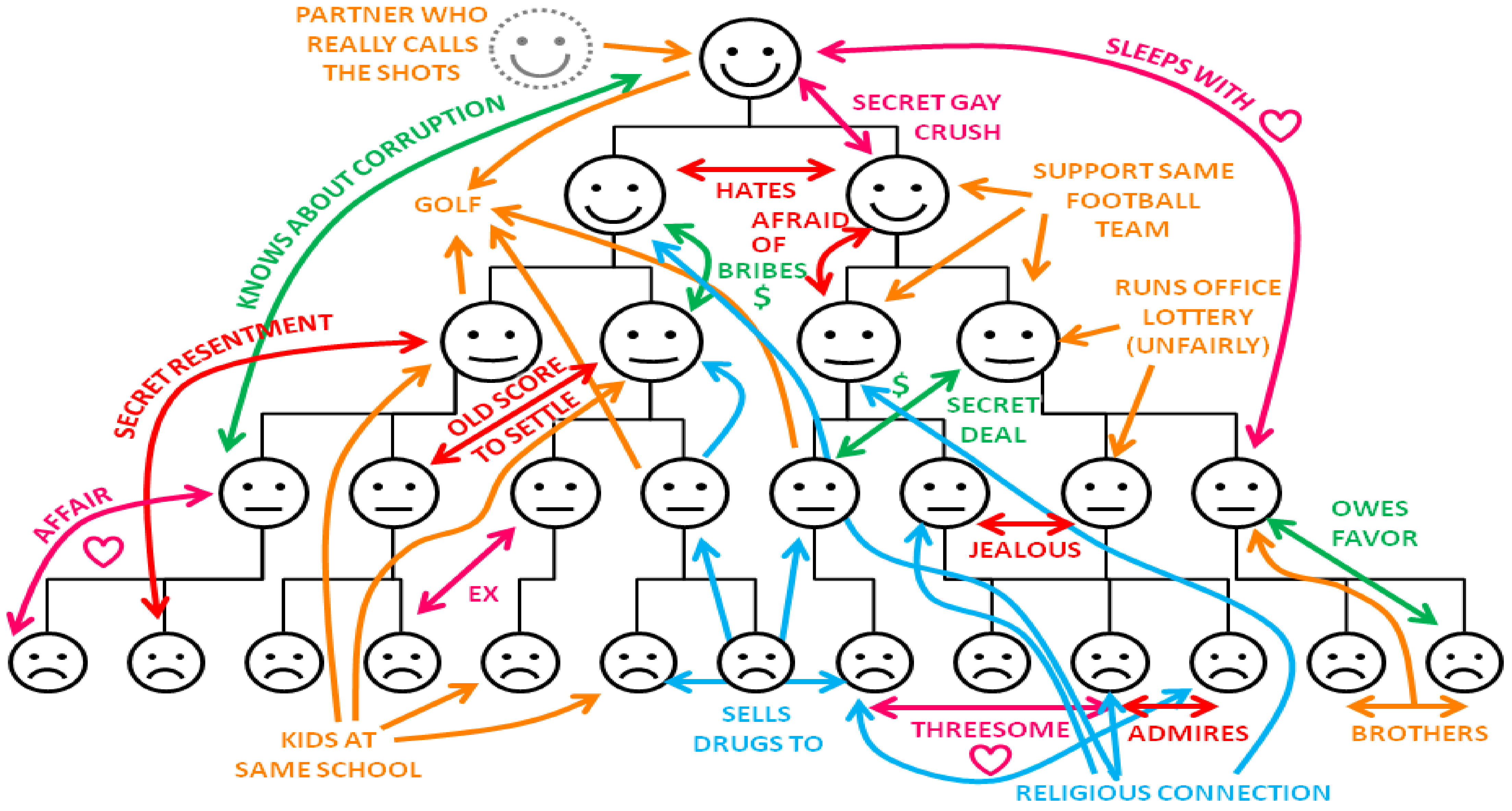

WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
scottadams@aol.com
www.dilbert.com
11-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.

# Scrum

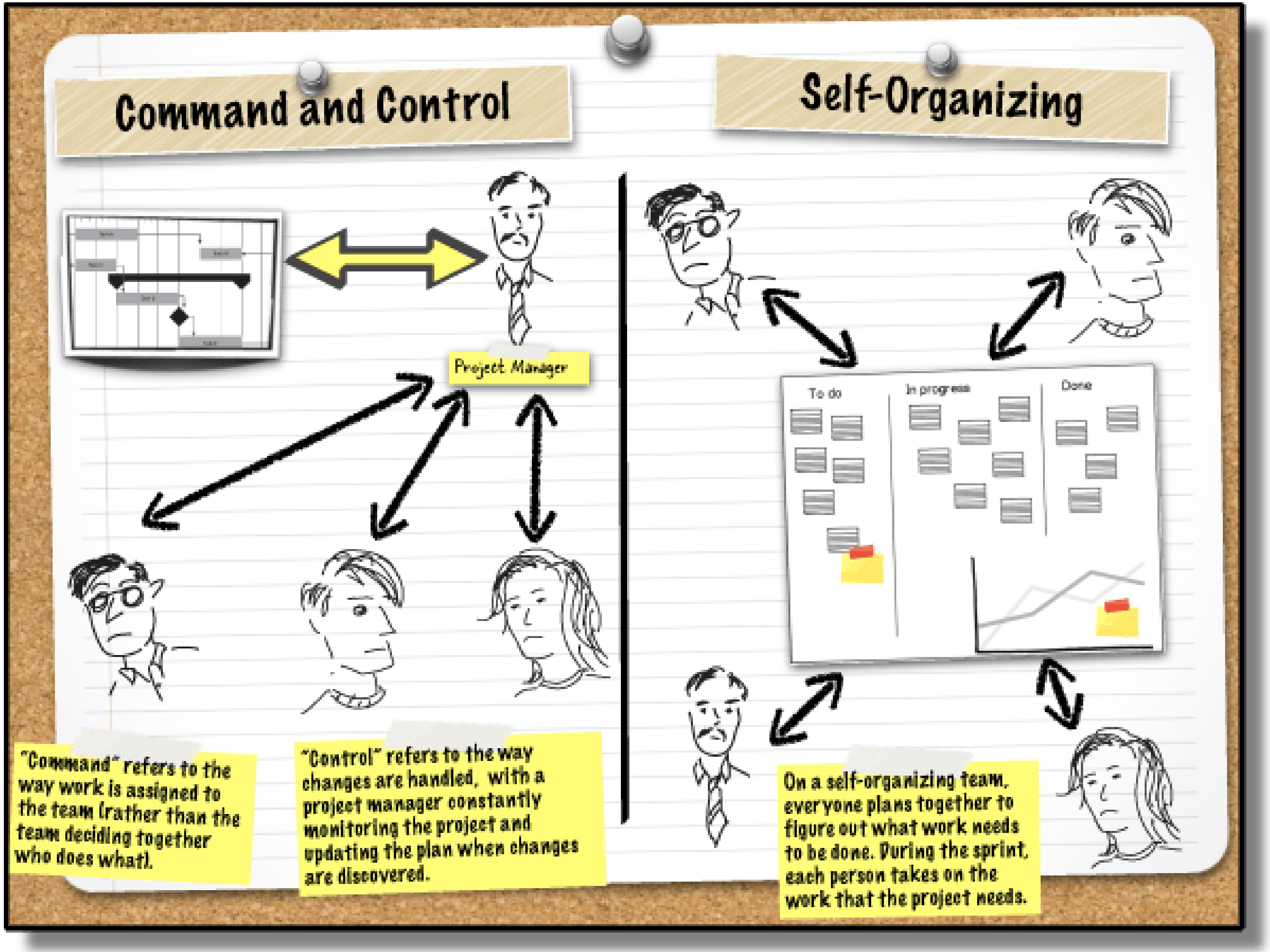
- Very popular implementation of Agile vision
- Emphasizes:
  - Team self management
  - Empirical feedback
  - Properly tested product increments
  - Short iterations

# REAL ORGANIZATION CHART

INTEGRATION TRAINING
www.integrationtraining.co.uk



# Teams are Self-Organizing



"Command" refers to the way work is assigned to the team (rather than the team deciding together who does what).

"Control" refers to the way changes are handled, with a project manager constantly monitoring the project and updating the plan when changes are discovered.

On a self-organizing team, everyone plans together to figure out what work needs to be done. During the sprint, each person takes on the work that the project needs.

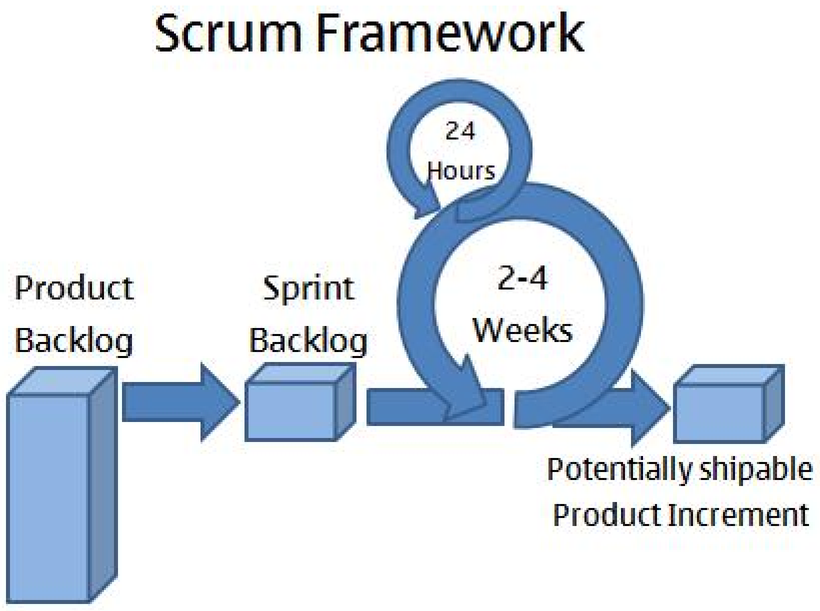
credit:
Self organizing teams
http://mariolucero.cl

# Scrum Details

- There is no project manager, responsibilities split between these roles:
  - Product Owner – Cares about what is made
  - Team – Self-organizes to create what is needed
  - Scrum Master – Removes blocks, ensures team has what they need
- Scrum has five meetings:
  - Backlog grooming
  - Spring planning
  - Daily scrum (15-minute standup)
  - Spring review
  - Sprint retrospective

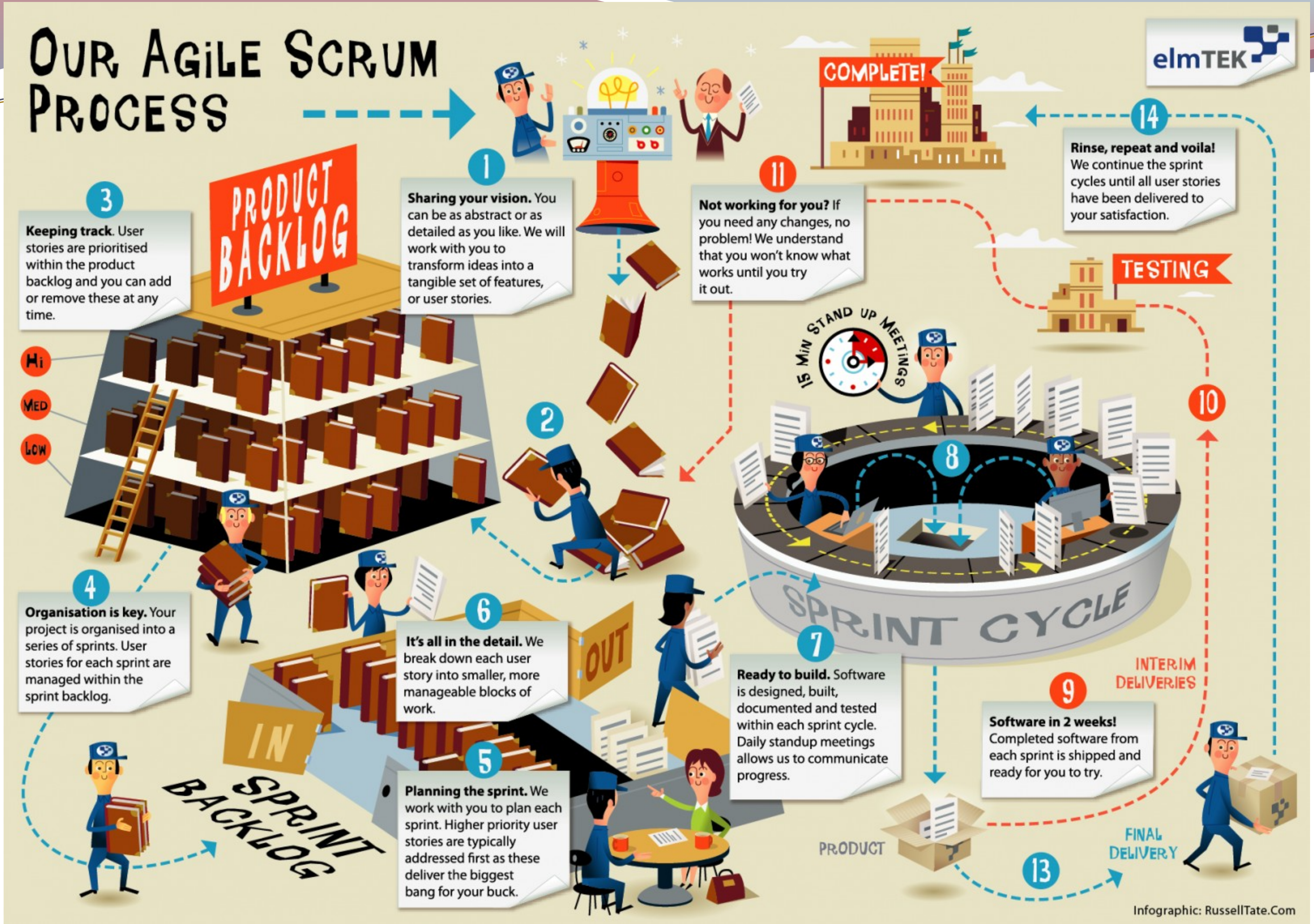
# Scrum Sprint

- Short duration
  - Usually 2-3 weeks in length
- Timeframe is fixed – output is variable
- Every iteration creates a workable version of the product
- Essentially the steps in waterfall process are repeated in each sprint
  - Mistakes have smaller impact and are found more quickly



Credit: [http://www.expertprogrammanagement.com/2010/08/the-scrum-process/](http://www.expertprogrammanagement.com/2010/08/the-scrum-process/)

# OUR AGILE SCRUM PROCESS



1. **Sharing your vision.** You can be as abstract or as detailed as you like. We will work with you to transform ideas into a tangible set of features, or user stories.

3. **Keeping track.** User stories are prioritised within the product backlog and you can add or remove these at any time.

4. **Organisation is key.** Your project is organised into a series of sprints. User stories for each sprint are managed within the sprint backlog.

5. **Planning the sprint.** We work with you to plan each sprint. Higher priority user stories are typically addressed first as these deliver the biggest bang for your buck.

6. **It's all in the detail.** We break down each user story into smaller, more manageable blocks of work.

7. **Ready to build.** Software is designed, built, documented and tested within each sprint cycle. Daily standup meetings allows us to communicate progress.

9. **Software in 2 weeks!** Completed software from each sprint is shipped and ready for you to try.

11. **Not working for you?** If you need any changes, no problem! We understand that you won't know what works until you try it out.

14. **Rinse, repeat and voila!** We continue the sprint cycles until all user stories have been delivered to your satisfaction.

Infographic: RussellTate.Com

# Government Case Example

Impact of Waterfall and Agile in Government IT

# Government Approach to Software

- Software as something that is procured:
  - Pencils, Rocket Launchers, Paper, Battleships, Dams, Highways, etc.
- Government prefers waterfall because:
  - Provides accountable command and control of software development
  - Relieves managers of the process from having to understand or be involved in technical decisions
  - Results in a final product that the government can buy
  - “Technical solutions are not the real issue, we will leave them to the software vendor to figure out”

# Government Approach to Software

| Perceived benefit of Waterfall: | … but Waterfall doesn’t deliver: |
|---|---|
| Provides accountable command and control of software development | Won’t help create a better product but will help when the report has to be written on why the process created software that didn’t meet expectations. |
| Relieves managers of the process from having to understand or be involved in technical decisions | Once the managers start to use and understand the finished software then they will see which technical decisions were incorrect. |
| Results in a final product that the government can buy | But more changes will be needed, the product is never finished. How many software systems do you know that don’t ever need to be updated? |
| “Technical solutions are not the real issue, we will leave them to the software vendor to figure out” | But details do matter, we know the “devil is in the details”, and good managers are able to get to the bottom of problems no matter how detailed. |

# Case Example

- HealthCare.gov website creation:
  - Recent and political example of government failure
- Cost went out of control:
  - Original budget: $94 million
  - Final cost: $1.7 billion
- Failure to launch:
  - Only 1% of interested people were able to enroll in first week
  - Stress tests (completed one day before launch!) showed that system could only handle 1,100 simultaneous users, needed to handle 50-60 thousand

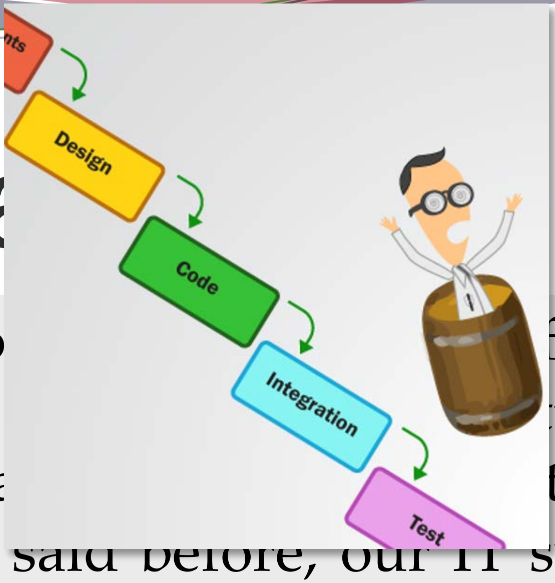
# Succinct Analysis from Leadership

And I think it’s fair to say that we have a pretty good track record of working with folks on technology and IT from our campaign where, both in 2008 and 2012, we did a pretty darn good job on that. So it’s not -- the idea that somehow we didn't have access or were interested in people’s ideas, I think isn’t accurate. What is true is that, as I said before, our IT systems, how we purchase technology in the federal government is cumbersome, complicated, and outdated.

And so this isn’t a situation where on my campaign I could simply say, who are the best folks out there; let’s get them around a table, let’s figure out what we’re doing, and we’re just going to continue to improve it and refine it and work on our goals. If you're doing it at the federal government level, you're going through 40 pages of specs and this and that and the other, and there are all kinds of laws involved, and it makes it more difficult. It's part of the reason why, chronically, federal IT programs are over budget, behind schedule.

President Barack Obama - November 14, 2013

# Succinct Analysis from Lea

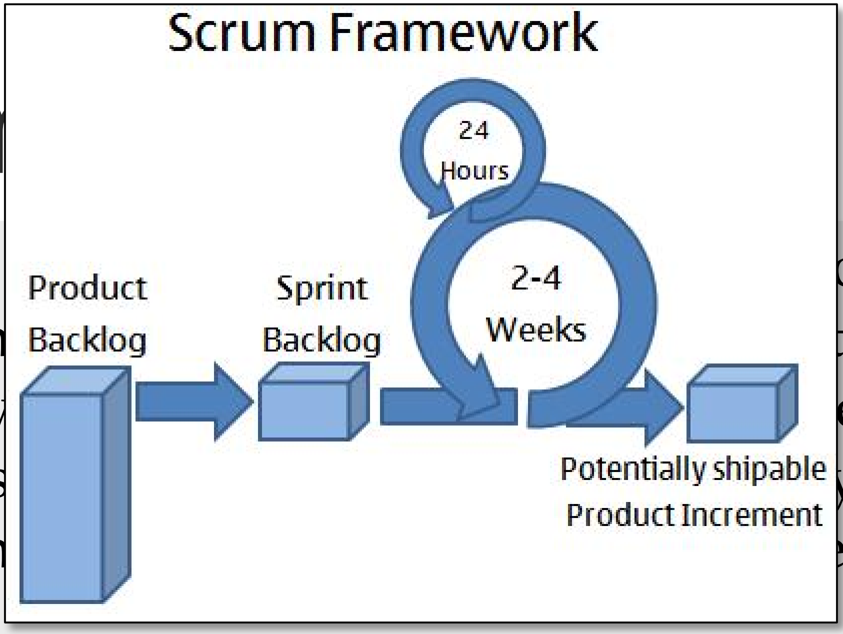


And I think it’s fair to say that we have a pretty good track reco folks on technology and IT from our campaign where, both in 2008 and ty darn good job on that. So it’s not -- the idea that somehow we didn't ha terested in people’s ideas, I think isn’t accurate. What is true is that, as I said before, our IT systems, how we purchase technology in the federal government is cumbersome, complicated, and outdated.

And so this isn’t a situation where on my campaign I could simply say, who are the best folks out there; let’s get them around a table, let’s figure out what we’re doing, and we’re just going to continue to improve it and refine it and work on our goals. If you're doing it at the federal government level, you're going through 40 pages of specs and this and that and the other, and there are all kinds of laws involved, and it makes it more difficult. It's part of the reason why, chronically, federal IT programs are over budget, behind schedule.

President Barack Obama - November 14, 2013

# Succinct Analysis fron



And I think it’s fair to say that we have a pretty good olks on
technology and IT from our campaign where, both in y darn
good job on that. So it’s not -- the idea that somehow w erested
in people’s ideas, I think isn’t accurate. What is true is ystems,
how we purchase technology in the federal governm ed, and
outdated.

And so this isn’t a situation where on my campaign I could simply say, who are the best folks out there; let’s get them around a table, let’s figure out what we’re doing, and we’re just going to continue to improve it and refine it and work on our goals. If you're doing it at the federal government level, you're going through 40 pages of specs and this and that and the other, and there are all kinds of laws involved, and it makes it more difficult. It's part of the reason why, chronically, federal IT programs are over budget, behind schedule.

President Barack Obama - November 14, 2013

# Agile is More Than a Label

- There were many reasons why HealthCare.gov failed, we are not going to cover them today
- However, there is one aspect of the project that should be noted here:
  - The project used Agile terms such as “sprints” and “story cards”
  - Many pundits assumed project was thus following Agile process
  - But testing was scheduled just before go-live, which model does this look like?
- If your project has “three architecture sprints, six coding sprints, followed by two test sprints” then this is probably Waterfall disguised as Agile
- Further reading: http://www.cio.com/article/2380827/developer/developer-6-software-development-lessons-from-healthcare-gov-s-failed-launch.html

# Wrap Up and Take Away

- Focus on software process not software products
  - A good process will produce good products
- Learn about Agile concepts
  - Begin to apply where you can
  - Advocate for changes in procurement that will support Agile
- Increase your technical comprehension
  - Effective managers know that details matter
  - Recognize your knowledge gaps
  - Be ready to learn new concepts

# QUESTIONS, COMMENTS, DISCUSSION

# THANKS SO MUCH!