

# Are Your Developers Bull\$h!t!ng You?

*And why that's the wrong question*

Bonnie Aumann, Algorithmics Inc.  
Philly Emerging Technologies for the Enterprise, April 2010

# Algorithmics

- Financial Services Software Vendor
- Risk Management
- Banks in the Fed 16, Multinationals

# The Story Behind the Talk

- Manage projects in Java, RoR, Clojure, AMQP

# The Story Behind the Talk

- Manage projects in Java, RoR, Clojure, AMQP
- I don't code

# The Story Behind the Talk

- Manage projects in Java, RoR, Clojure, AMQP
- I don't code
- Psych Degree

# The Story Behind the Talk

- Manage projects in Java, RoR, Clojure, AMQP
- I don't code
- Psych Degree
- Geocities! </blink>

THE EXPERT'S VOICE™



J. Hank Rainwater

*Foreword by Tom Mochal,  
respected project management guru  
and founder of [www.TenStep.com](http://www.TenStep.com)*

*Graphics by Melanie Wells*

# **Herding Cats:** **A Primer for Programmers Who Lead Programmers**

Guide your programming team to success by becoming a *leader*, not just a manager



Identify and overcome obstacles before they impact you or your team



Learn from an author with over 30 years of experience in science, engineering,  
programming, and the management of technical teams

**a!**  
Apress™

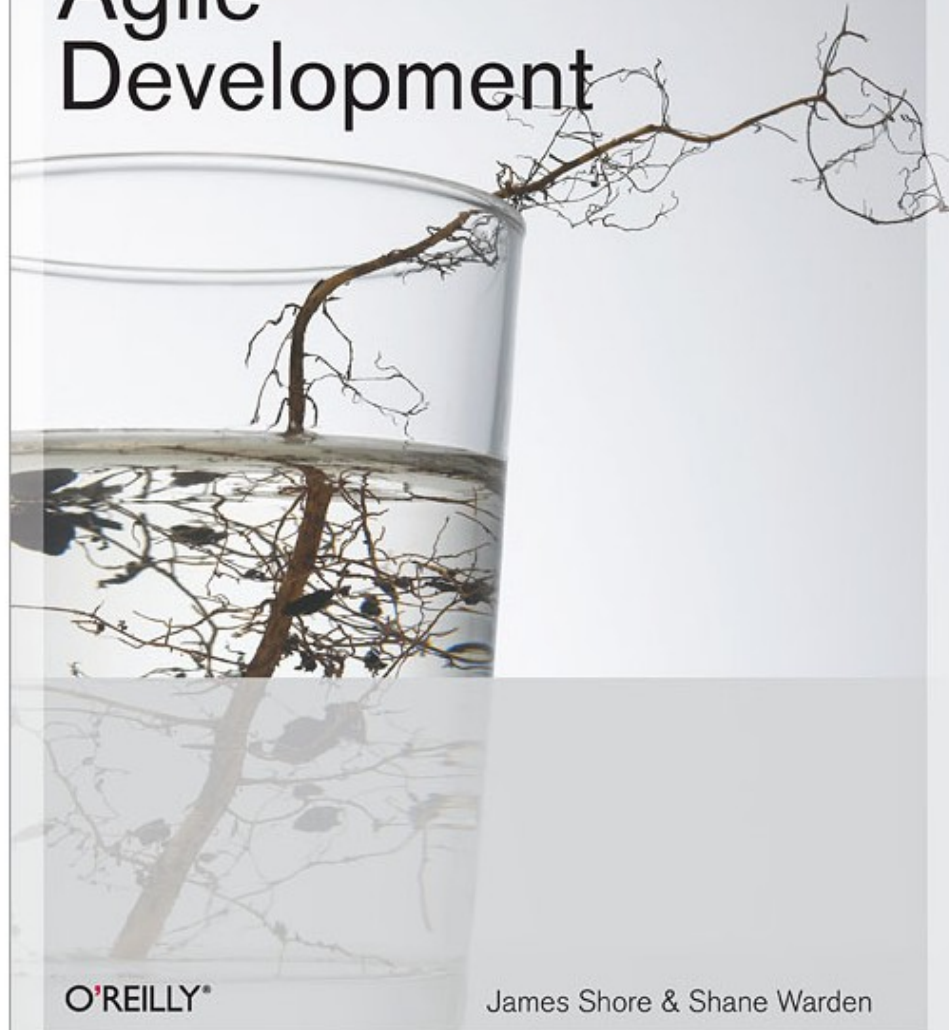
What?



The Art of

/THEORY/IN/PRACTICE

# Agile Development



O'REILLY®

James Shore & Shane Warden

# The Need Behind the Question

Dependent on team for success

# The Need Behind the Question

Dependent on team for success

Must evaluate trustworthiness

# The Need Behind the Question

Dependent on team for success

Must evaluate trustworthiness

Amplified by specialized knowledge imbalance

# The Need Behind the Question

Dependent on team for success

Must evaluate trustworthiness

Amplified by specialized knowledge imbalance

Devs ask too

# Blame Culture

Suspicion and lack of trust

# Blame Culture

Suspicion and lack of trust

Toxic to success

# Blame Culture

Suspicion and lack of trust

Toxic to success

*Our questions form our culture*



# Our Job

Deliver value to our customers

# Our Job

Deliver value to our customers

Maximize efficiency and productivity

# Our Job

Deliver value to our customers

Maximize efficiency and productivity

You can't build it all yourself

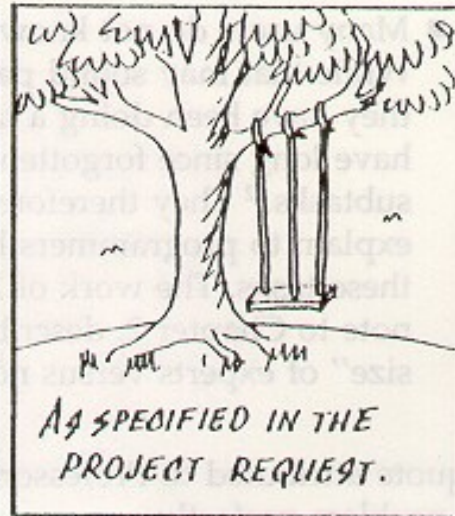
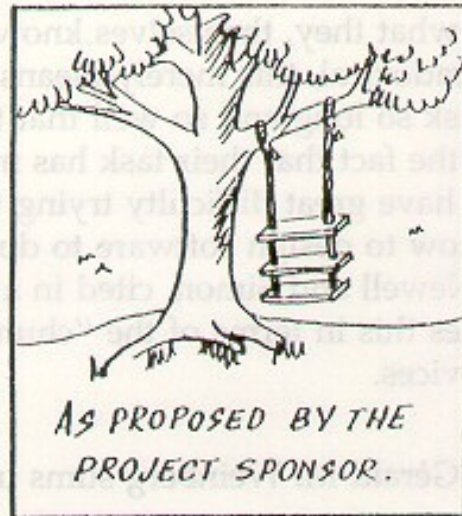
## Reframe the Question

~~Are my developers  
bull\$h!tting?~~

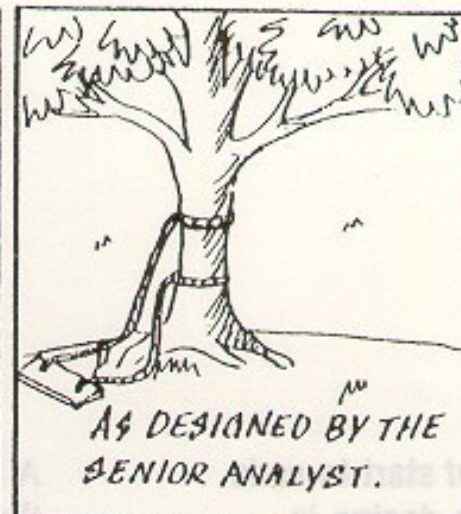
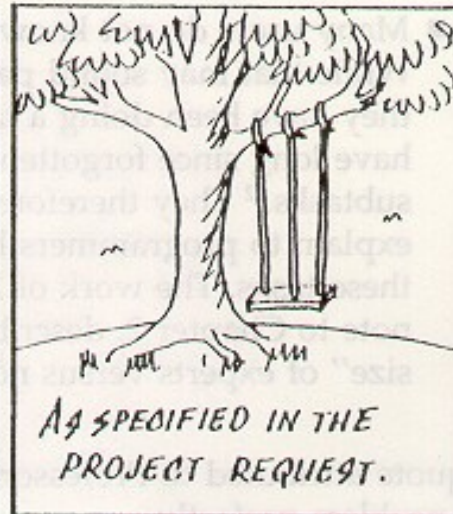
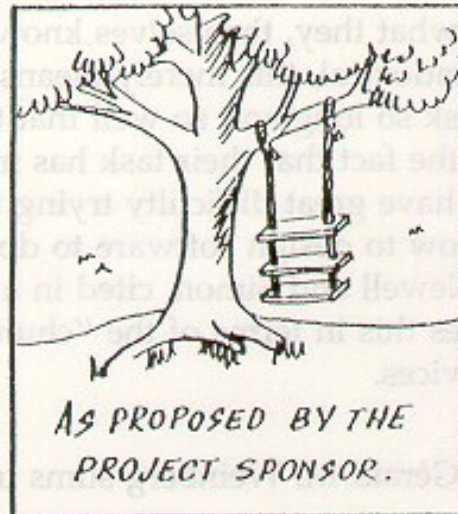
What do I need in order to  
trust my team?



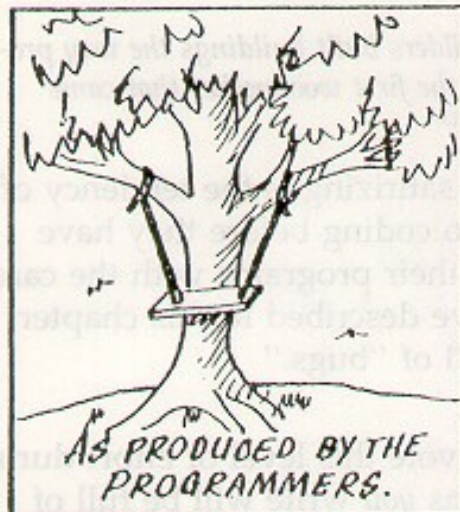
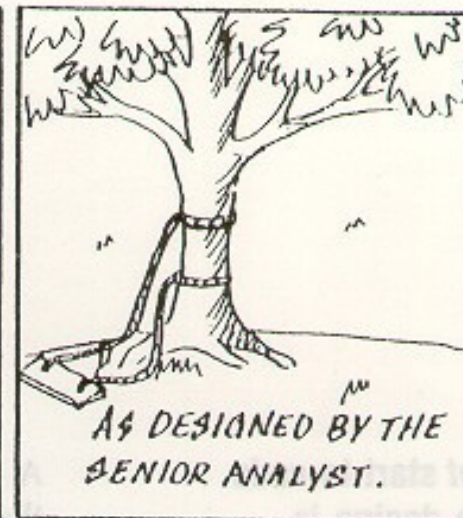
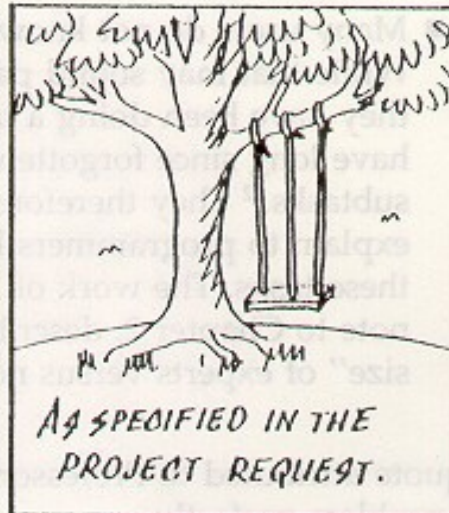
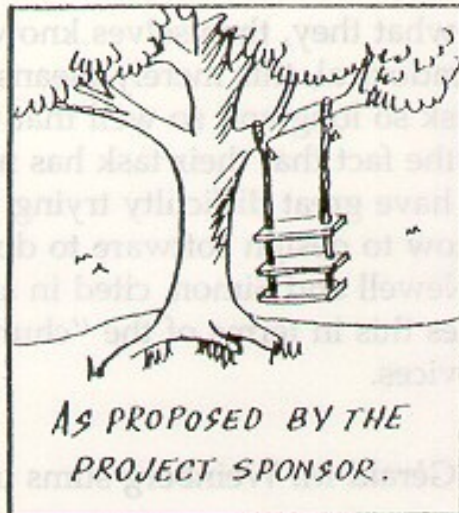
As proposed by the project sponsor.



As specified in the project request.

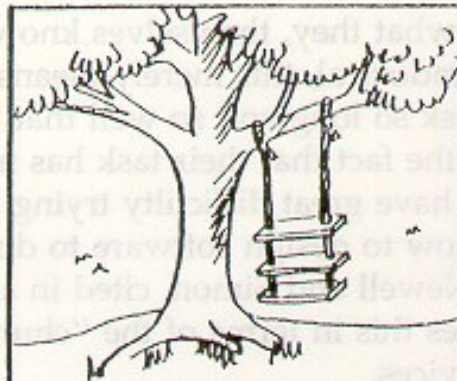


As designed by the senior analyst.

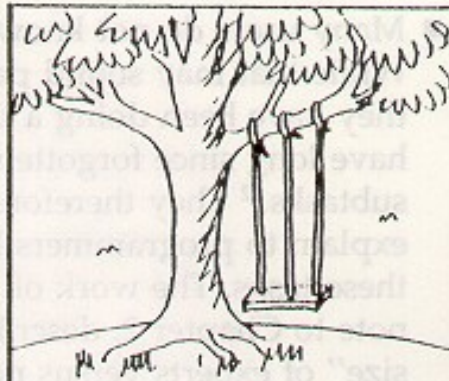


As produced by the programmers.

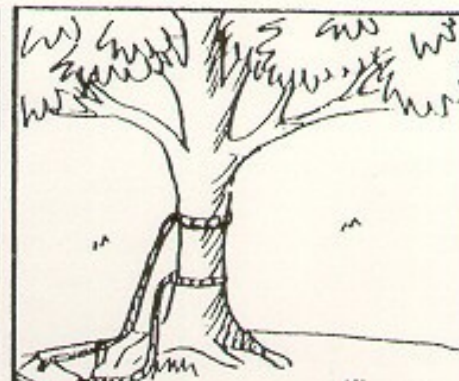




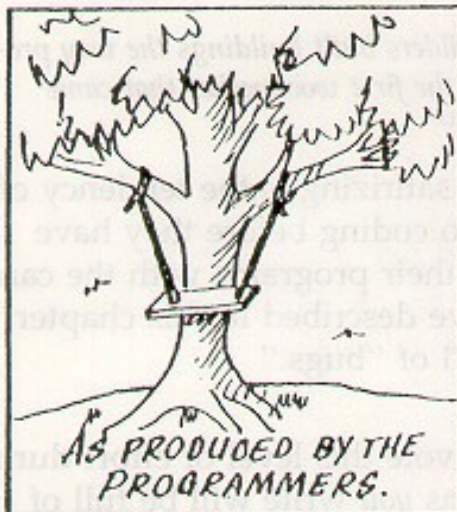
AS PROPOSED BY THE  
PROJECT SPONSOR.



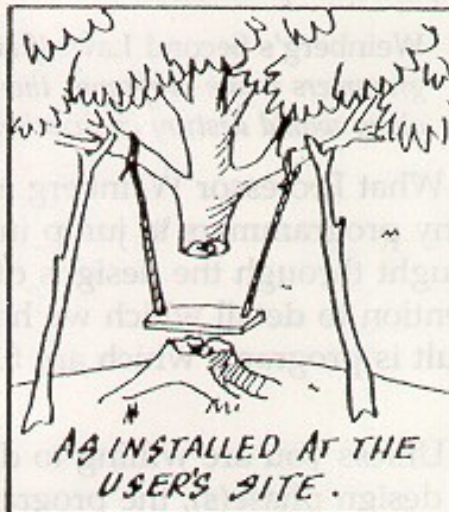
AS SPECIFIED IN THE  
PROJECT REQUEST.



AS DESIGNED BY THE  
SENIOR ANALYST.

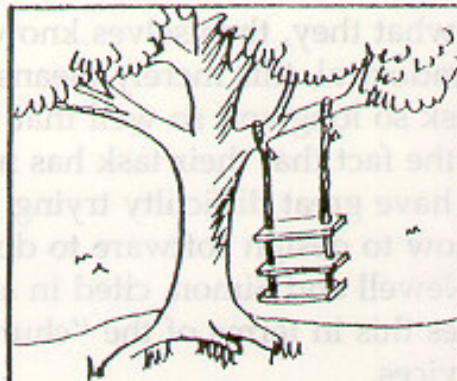


AS PRODUCED BY THE  
PROGRAMMERS.

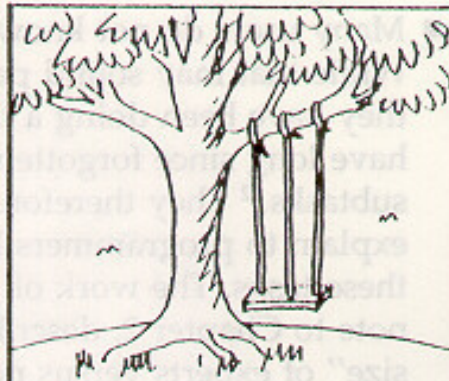


AS INSTALLED AT THE  
USER'S SITE.

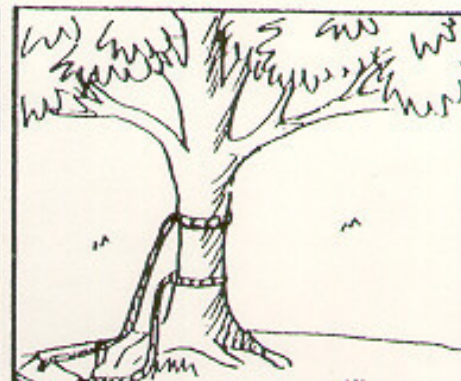
As installed at the user's site.



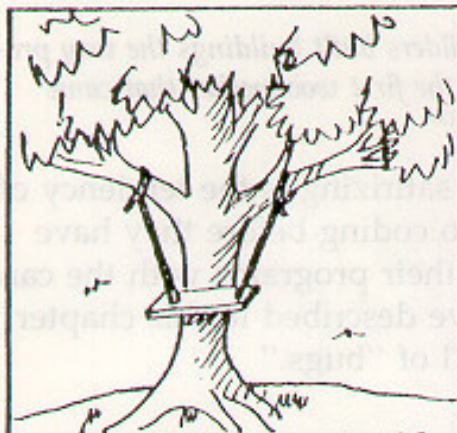
AS PROPOSED BY THE  
PROJECT SPONSOR.



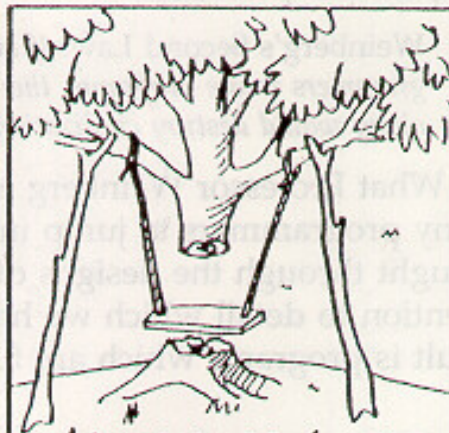
AS SPECIFIED IN THE  
PROJECT REQUEST.



AS DESIGNED BY THE  
SENIOR ANALYST.



AS PRODUCED BY THE  
PROGRAMMERS.



AS INSTALLED AT THE  
USER'S SITE.



WHAT THE USER WANTED.

What the user actually wanted.





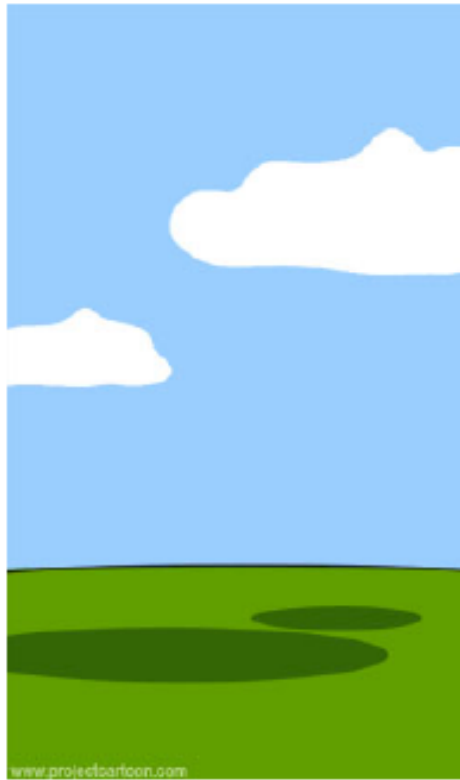
How the business  
consultant described it

The "Tree Cartoon" is still  
alive and well  
in Web 2.0

<http://projectcartoon.com/>



How the business  
consultant described it

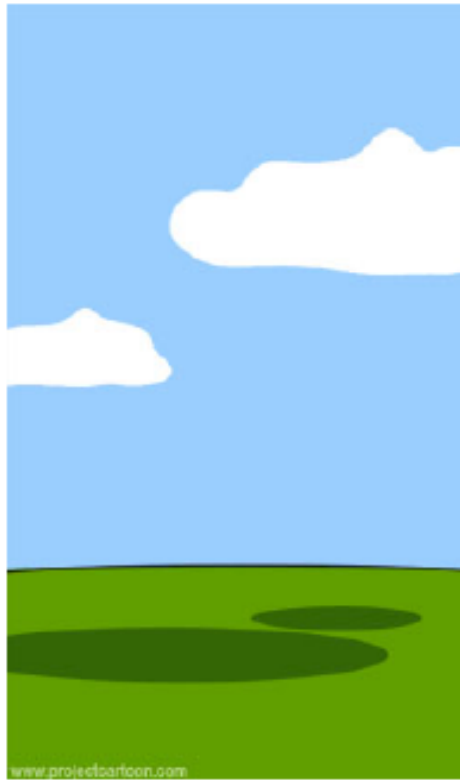


How the project was  
documented

<http://projectcartoon.com/>



www.projectcartoon.com  
How the business  
consultant described it



www.projectcartoon.com  
How the project was  
documented



www.projectcartoon.com  
The disaster recovery  
plan

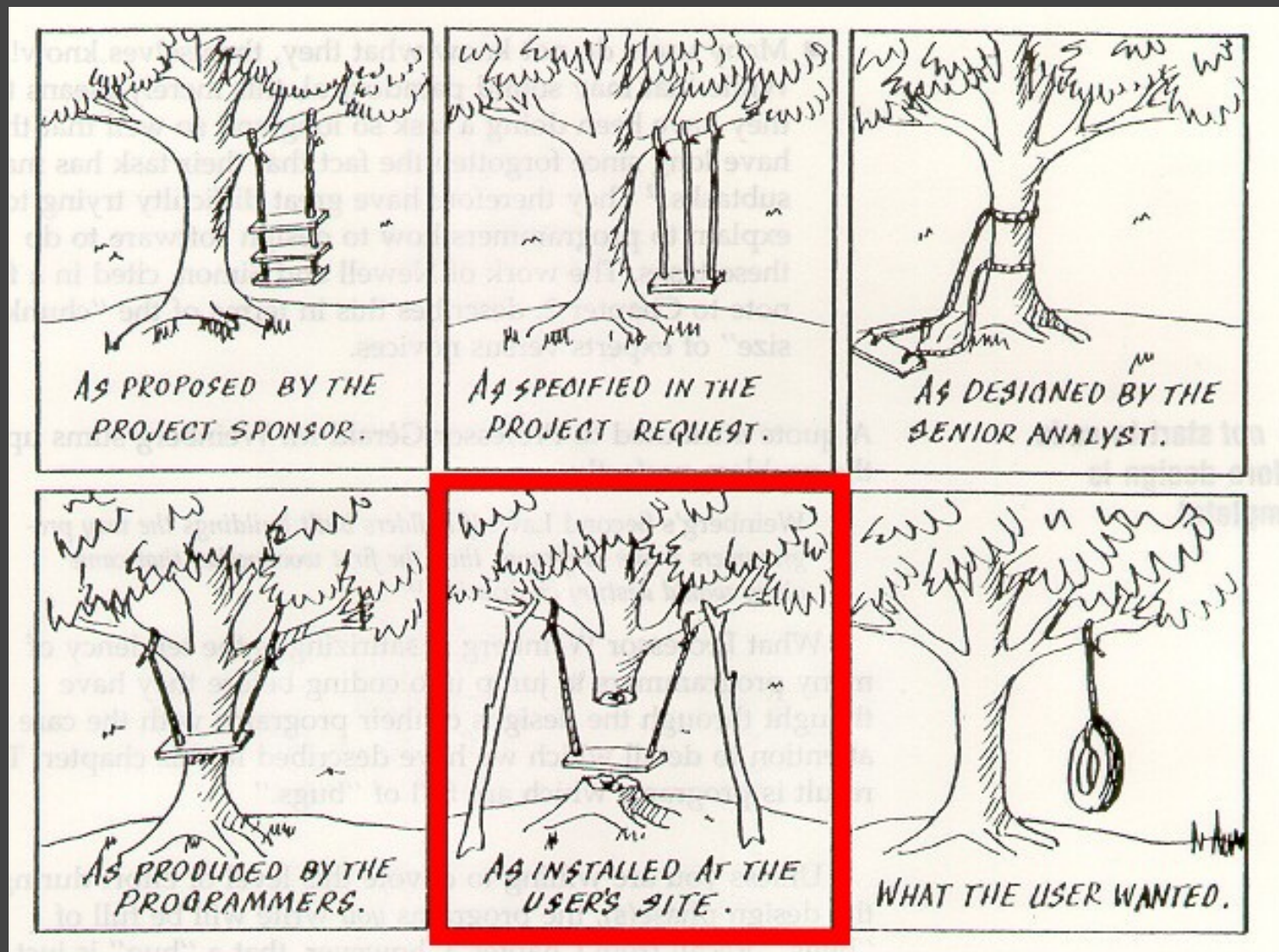
<http://projectcartoon.com/>

# Project Management as Risk Mitigation





# Our Deepest Fear



So how do we get here...



To what they actually wanted.



# Risk Mitigation through Authority

- Requirements. **Requirements. Requirements!**

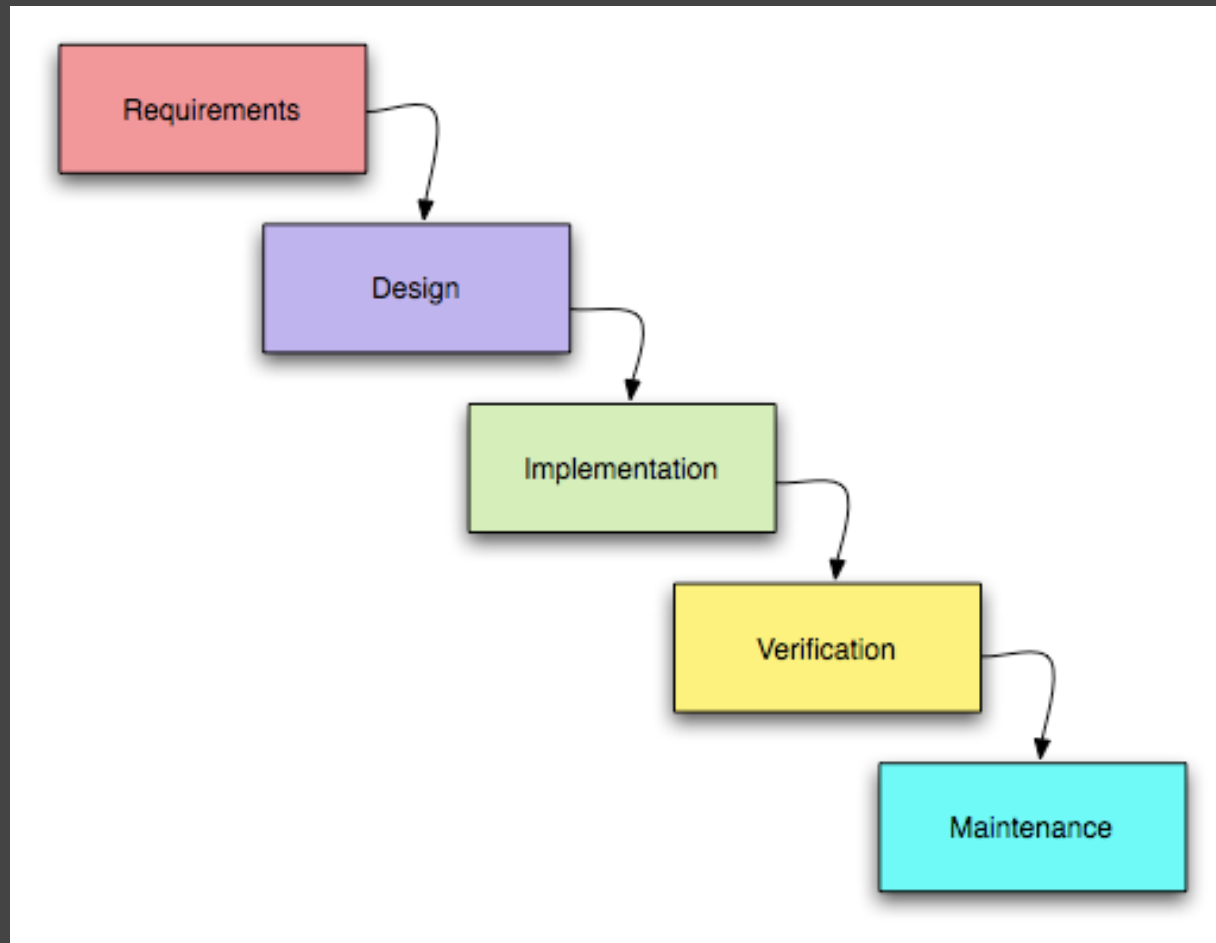
# Risk Mitigation through Authority

- Requirements. **Requirements. Requirements!**
- A Plan *aka* Master Roadmap
  - (based on estimates by non-developers)

# Risk Mitigation through Authority

- Requirements. **Requirements. Requirements!**
- A Plan *aka* Master Roadmap
  - (based on estimates by non-developers)
- Sign-offs
  - It's *your* @\$\$ on the line

# Risk Mitigation through Authority



Simplified Waterfall Model

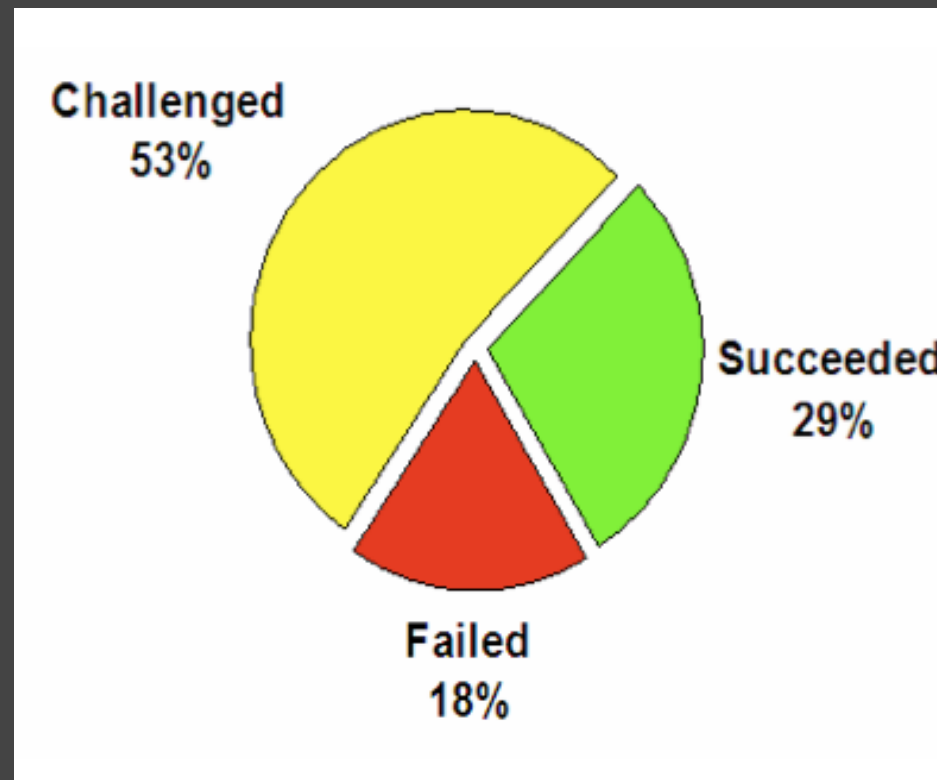


# And yet...

The first formal description  
of the waterfall model  
(though not called by that name),  
was presented as  
***an example of a flawed,  
non-working model***  
(Royce 1970) (from Wikipedia)

# Illusion of Control - Madness in the Method

2004 CHAOS Report (Standish Group)



# Illusion of Control - Detailed Requirements

“

*With the Waterfall approach,  
a great idea,  
late in the development cycle  
is **not a gift**, it's a **threat**.*

*Pete Deemer, Chief Product Officer, Yahoo! India  
Research and Development  
and*

*Gabrielle Benefield, Senior Director of Agile Development,  
Yahoo!, Inc.*

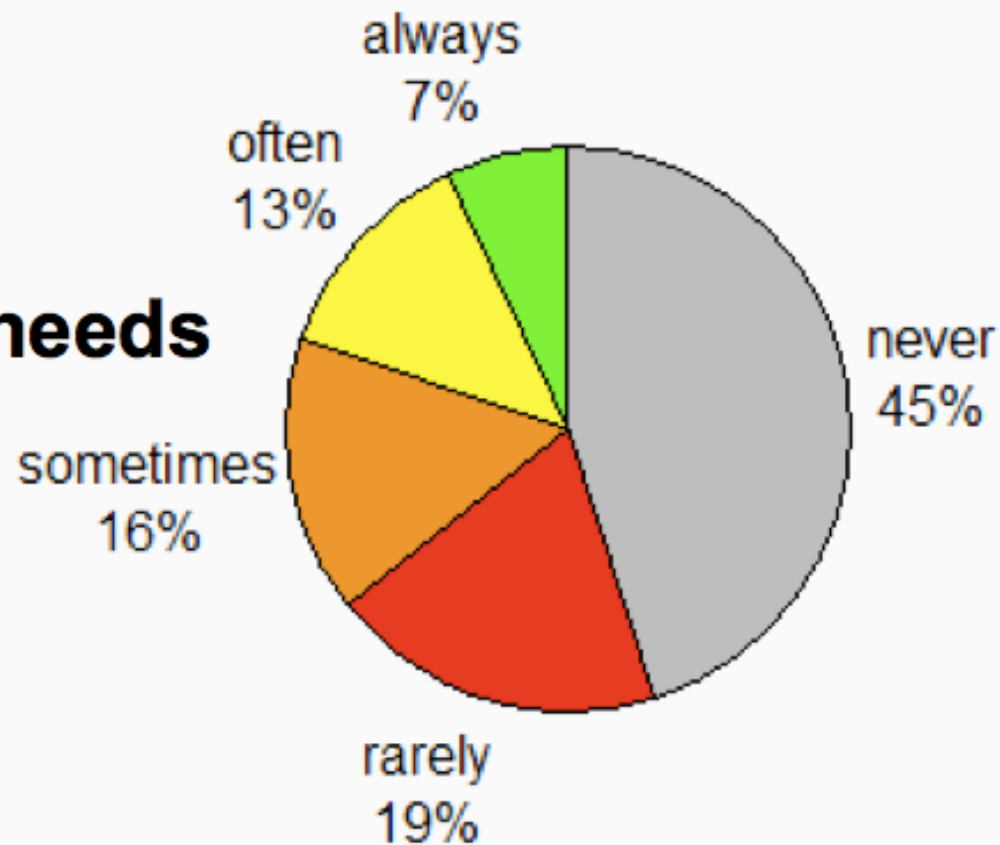
*Ref: Dr. Jeff Sutherland, Agile 2007*



# Illusion of Control - Detailed Requirements

80% of Software **Sucks**

## Fit to user needs



*Jim Johnson. The Standish Group International Inc. 2002*

# Illusion of Control - The Plan *is* the Problem

## Plan-Based Failure Rates (CHAOS report 2004: Standish group)

Project Size	People	Time (months)	Success Rate
Less than \$750k	6	6	55%
\$750k - \$1.5M	12	9	33%
\$1.5M - \$3M	25	12	25%
\$3M - \$6M	40	18	15%
\$6M - 10M	+250	+24	8%
Over \$10M	+500	36	0

# Illusion of Control - The Plan *is* the Problem

“

We deliver on time and on budget 100% of the time. Our waterfall projects are 100% successful. The customer then says it is not what they wanted 100% of the time.

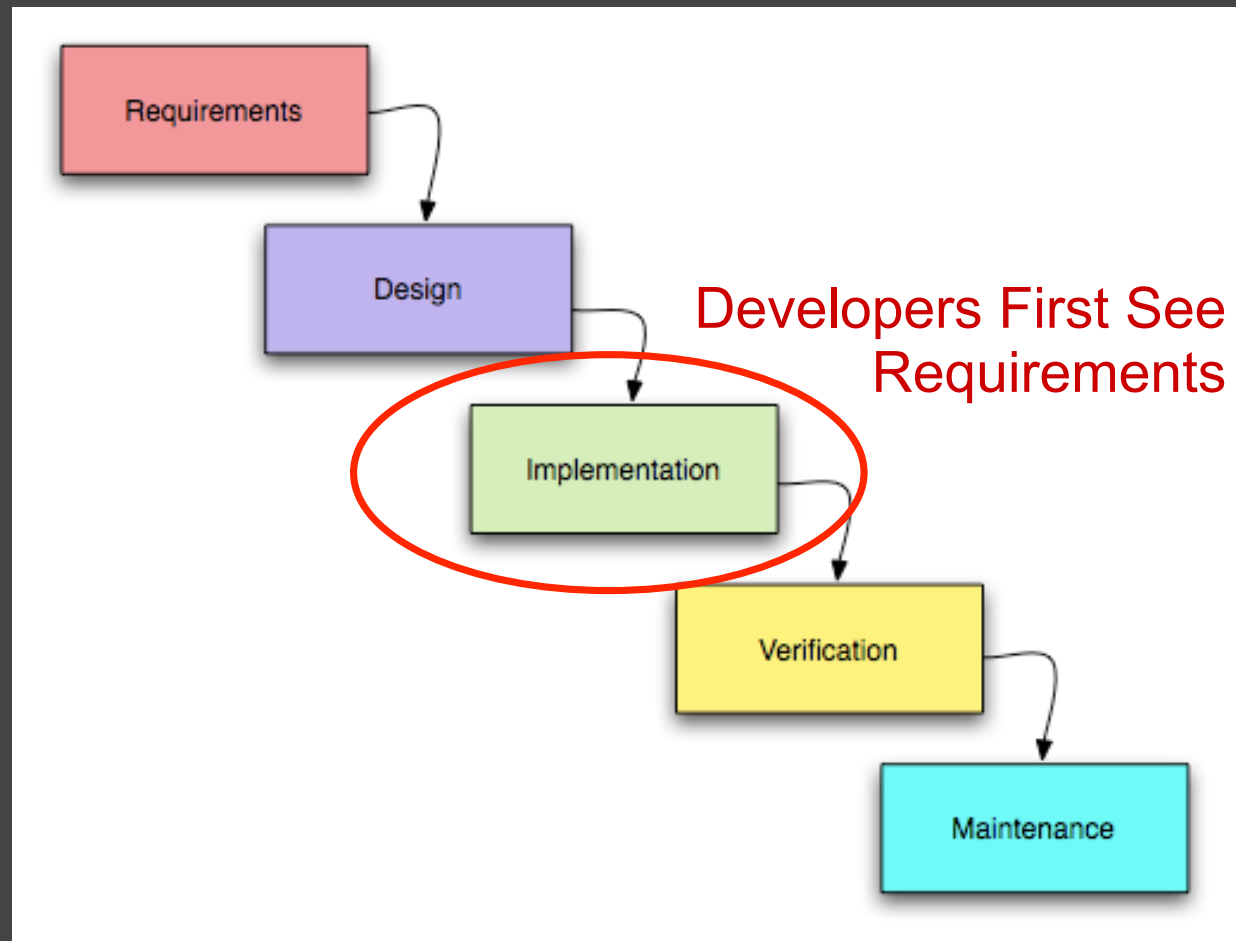
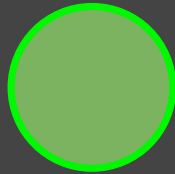
Our failure rate is 100%.

*BellSouth management*  
*-- A CMM Level 5 corporation*

*Ref: Dr. Jeff Sutherland, Agile 2007*

# Illusion of Control - The Plan *is* the Problem

Project Green-light



# Illusion of Control – Silos of Responsibility

“

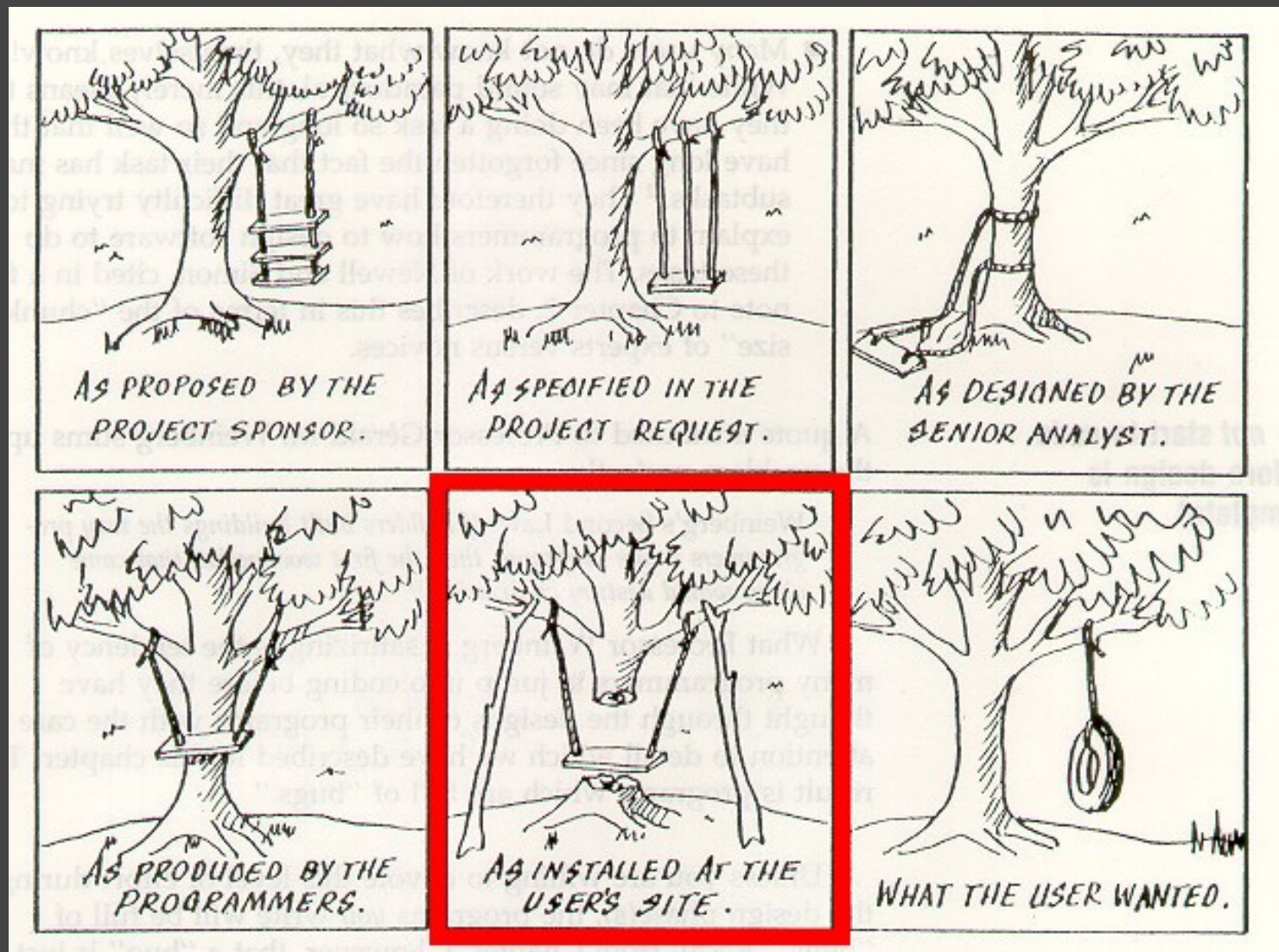
*[At Massachusetts General Hospital]  
Handoffs commonly lead to **patient harm**.*

"Flubbed handoffs of patients between doctors leads to errors, MGH survey finds."

Elizabeth Cooney September 23, 2008.

[http://www.boston.com/news/health/blog/2008/09/hazards\\_of\\_hand.html](http://www.boston.com/news/health/blog/2008/09/hazards_of_hand.html)

# Our Deepest Million-Dollar Fear



So what do we do?

The first rule of holes:  
When you find yourself in  
one, stop digging.

@estherderby

<http://twitter.com/estherderby/status/11752553863>



# Risk Mitigation: An Agile Alternative

Derail the dynamic

Ignorance ➡ miscommunication ➡ mistrust

# Risk Mitigation: An Agile Alternative

Derail the dynamic

Ignorance  miscommunication  mistrust

Ask what makes it easier...

- for trust to develop between a manager and their team

# Risk Mitigation: An Agile Alternative

Derail the dynamic

Ignorance  miscommunication  mistrust

Ask what makes it easier...

- for trust to develop between a manager and their team
- for a manager to evaluate trustworthiness separately from competence

# Risk Mitigation: An Agile Alternative

Derail the dynamic

Ignorance  miscommunication  mistrust

Ask what makes it easier...

- for trust to develop between a manager and their team
- for a manager to evaluate trustworthiness separately from competence
- for developers to appreciate that managers see demonstrating competence and demonstrating trustworthiness as different things

# Risk Mitigation: An Agile Alternative

Derail the dynamic

Ignorance  miscommunication  mistrust

Ask what makes it easier...

- for trust to develop between a manager and their team
- for a manager to evaluate trustworthiness separately from competence
- for developers to appreciate that managers see demonstrating competence and demonstrating trustworthiness as different things
- for developers and managers to believe that honesty will bring about better results than dishonesty

Risk Mitigation through Collaboration

# Inspect and Adapt

Take trust out of the equation

Let it grow from a foundation of small wins

# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front

# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front
- Rapid feedback and response
  - vs. A Master Roadmap



# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front
- Rapid feedback and response
  - vs. A Master Roadmap
- Whole team
  - vs. Sign-offs

# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front

# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front

Just-In-Time requirements  
direct collaboration btw customers and developers

# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front

Just-In-Time requirements  
direct collaboration btw customers and developers

Frequent releases

# Risk Mitigation through Collaboration

- Incremental design
  - vs. Detailed requirements up front

Just-In-Time requirements

direct collaboration btw customers and developers

Frequent releases

Rapid feedback from the actual customer

# Risk Mitigation through Collaboration

- Rapid feedback and response
  - vs. A Master Roadmap

# Risk Mitigation through Collaboration

- Rapid feedback and response
  - vs. A Master Roadmap

Estimates made by developers  
and when they don't know, try something then measure

# Risk Mitigation through Collaboration

- Rapid feedback and response
  - vs. A Master Roadmap

Estimates made by developers  
and when they don't know, try something then measure

Prioritization by customers  
to ensure they get what they most want first



# Risk Mitigation through Collaboration

- Rapid feedback and response
  - vs. A Master Roadmap

Estimates made by developers  
and when they don't know, try something then measure

Prioritization by customers  
to ensure they get what they most want first

Burndown to commitment dates  
visible on walls to provide visibility

# Risk Mitigation through Collaboration

- Rapid feedback and response
  - vs. A Master Roadmap

Estimates made by developers  
and when they don't know, try something then measure

Prioritization by customers  
to ensure they get what they most want first

Burndown to commitment dates  
visible on walls to provide visibility

Retrospectives and kaizen  
empower teams self-organize improvements

# Risk Mitigation through Collaboration

- Whole team
  - vs. Sign-offs

# Risk Mitigation through Collaboration

- Whole team
  - vs. Sign-offs

Everyone Sit Together (Developers, testers, PMs, BAs, customers)

# Risk Mitigation through Collaboration

- Whole team
  - vs. Sign-offs

Everyone Sit Together (Developers, testers, PMs, BAs, customers)

Build empathy

Break down us vs. them mentality

# Risk Mitigation through Collaboration

- Whole team
  - vs. Sign-offs

Everyone Sit Together (Developers, testers, PMs, BAs, customers)

Build empathy

Break down us vs. them mentality

Osmotic Communication

“Cocktail party effect”

Locate people according to conversations they should overhear

*Ref: Art of Agile Development, <http://jamesshore.com/Agile-Book/sit-together.html>*

# High-Trust Teams are High-Performing Teams

IEEE field study of co-located teams

6 team pilot

# High-Trust Teams are High-Performing Teams

IEEE field study of co-located teams

6 team pilot

2x Productivity X 1/3 Time to Market

=

2/3 reduction in Time to Market

compared to that company's baseline development efforts



# High-Trust Teams are High-Performing Teams

IEEE field study of co-located teams

6 team pilot

2x Productivity X 1/3 Time to Market

=

2/3 reduction in Time to Market

compared to that company's baseline development efforts

*Later successfully repeated by 11 other teams, resulting in investment for 112 other such teams.*

*Ref: Art of Agile Development, <http://jamesshore.com/Agile-Book/sit-together.html>*

*What's a PM to do?*

Agile Project Management

*Servant Leadership*

# Agile Project Management

## *Servant Leadership*

✓ *Facilitation*

# Agile Project Management

## *Servant Leadership*

- ✓ *Facilitation*

- ✓ *Surface New Data*

# Agile Project Management

## *Servant Leadership*

- ✓ *Facilitation*
- ✓ *Surface New Data*
- ✓ *Smooth Interactions*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*



# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*
- *The traditional PM tries to control.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*
- *The traditional PM tries to control.*
- *The Facilitative PM adapts to reality.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*
- *The traditional PM tries to control.*
- *The Facilitative PM adapts to reality.*
- *The traditional PM attempts to force reality into their view of how things should be*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*
- *The traditional PM tries to control.*
- *The Facilitative PM adapts to reality.*
- *The traditional PM attempts to force reality into their view of how things should be*
- *We are not reactive, we are prepared.*

# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*
- *The traditional PM tries to control.*
- *The Facilitative PM adapts to reality.*
- *The traditional PM attempts to force reality into their view of how things should be*
- *We are not reactive, we are prepared.*
- *We are aware of our options but do not force reality to be what it isn't*



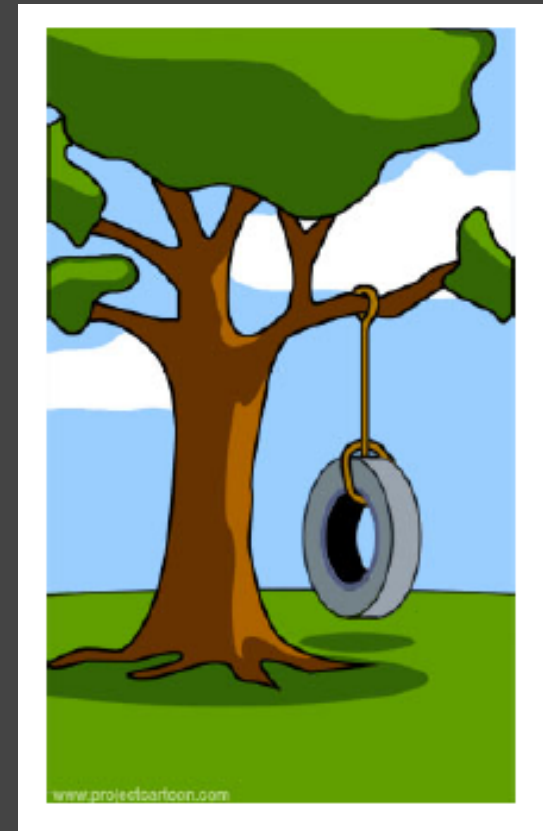
# Agile Project Management

- *The facilitative PM owns the responsibility for finding the best solution.*
- *We do not own the right to chose the best solution.*
- *Facilitative PM uses scenarios and feedback to create opportunities to adapt.*
- *Traditional uses restrictive controls so no variation occurs.*
- *The Facilitative PM responds to the situation.*
- *The traditional PM tries to control.*
- *The Facilitative PM adapts to reality.*
- *The traditional PM attempts to force reality into their view of how things should be*
- *We are not reactive, we are prepared.*
- *We are aware of our options but do not force reality to be what it isn't*
- *Rather than attempt to control the behavior of the system, we allow the behavior to emerge and then adapt*

# Agile Project Management

## *Eye on the Prize*

- ✓ *Deliver the most value to your customers*
- ✓ *Advise the business of its options*
- ✓ *Remove impediments from your team's productivity*
- ✓ *Forget to ask if you're developers are bull\$hitting you*



# Thank you!

*Thanks for listening!*  
*It's been an honor.*

# Contact me

Bonnie Aumann

[bonnie@bonniea.com](mailto:bonnie@bonniea.com)

@bonniea

*I'll also be an attendee at...*

**Lean Systems & Software Conference** in Atlanta



# Resources

*The Art of Agile Development*, 2008.

Text available free online <http://jamesshore.com/Agile-Book/>

*The Chaos Report Replicated*

A Replicated Survey of IT Software Project Failures. IEEE Magazine, September/October 2008 (vol. 25 no. 5).

# Image Credits

The dubious history of the tree swing is documented at <http://www.businessballs.com/treeswing>

Project Cartoon <http://projectcartoon.com/cartoon/72641>

Plate Twirler - unattributed

Waterfall Model

Originally from en.wikipedia; description page is/was here  
[http://en.wikipedia.org/w/index.php?title=Image  
%3AWaterfall\\_model.png](http://en.wikipedia.org/w/index.php?title=Image%3AWaterfall_model.png)

Admiral Ackbar

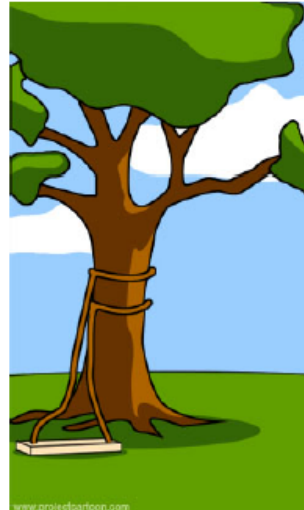
<http://pajamasmedia.com/eddriscoll/files/2009/10/ackbar.jpg>



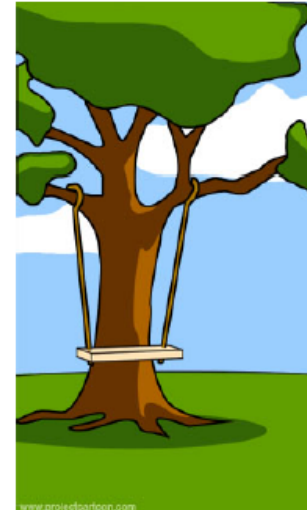
How the project sponsor explained it



How the business consultant described it



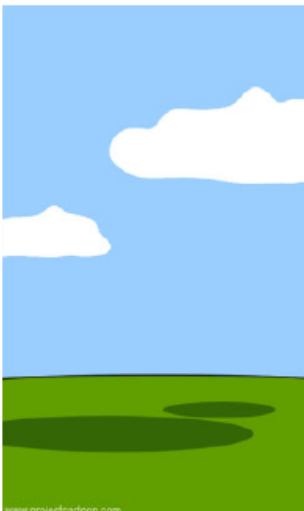
How the project leader understood it



How the programmers built it



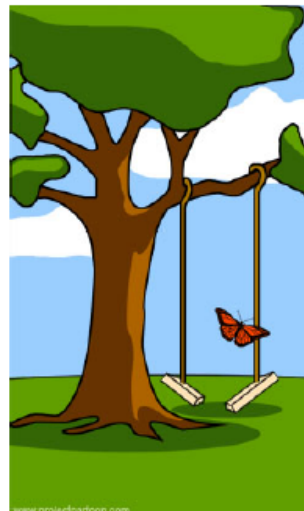
How operations installed it



How the project was documented



The disaster recovery plan



How it performed under load



What the customer really needed