

The 7 Deadly Sins of Almost Being Agile

Bob Hartman
Richard Lawrence



www.agilecooperative.com



Logistics

- Please turn cell phones, pagers, PDA's, etc to the “stun yourself” setting (rather than the “annoy everyone else” setting)
- We will take a 30 minute break after 1.5 hours
- Please ask questions when they come up rather than waiting
- Be prepared to participate! This presentation requires a lot of work on your part.

About us...



www.agilecooperative.com

About us...

About you...

**About this
session...**

Our Fictional Team

W-Agilists

Sally the Project Manager



Bill the Business Analyst

**CBAP[®] Certified Business
Analysis Professional**

Tom the Product Manager



4 Developers



2 Testers



The First Agile Project

6 months

Expected 25 features

Delivered 10 features

3 “most important” cut

Prior release 72 defects

Agile release 70 defects

Customers unhappy

The Second Agile Project

6 months

Expected 25 features

Delivered 8 features

5 “most important” cut

Prior release 70 defects

This release 79 defects

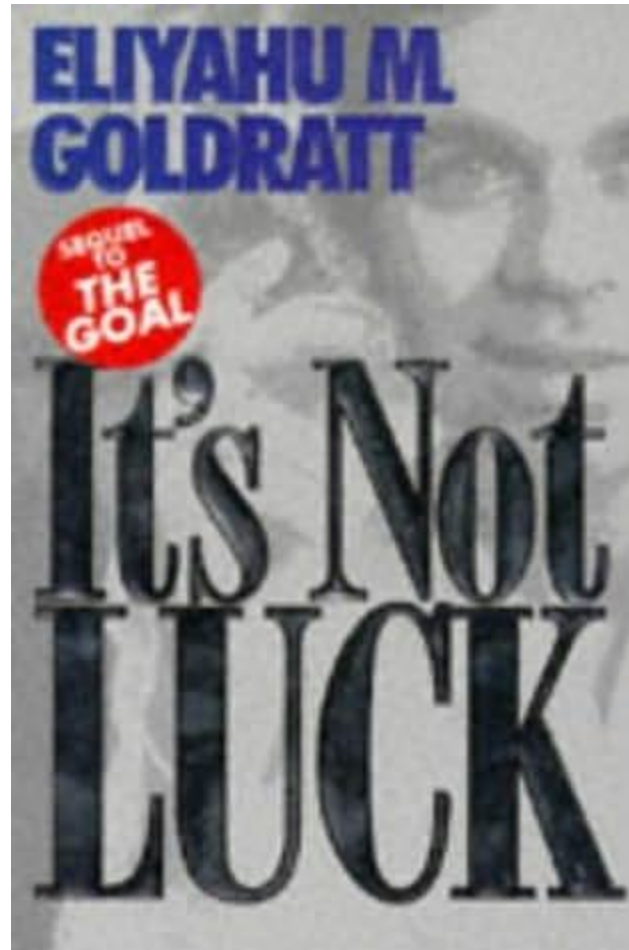
Customers VERY unhappy

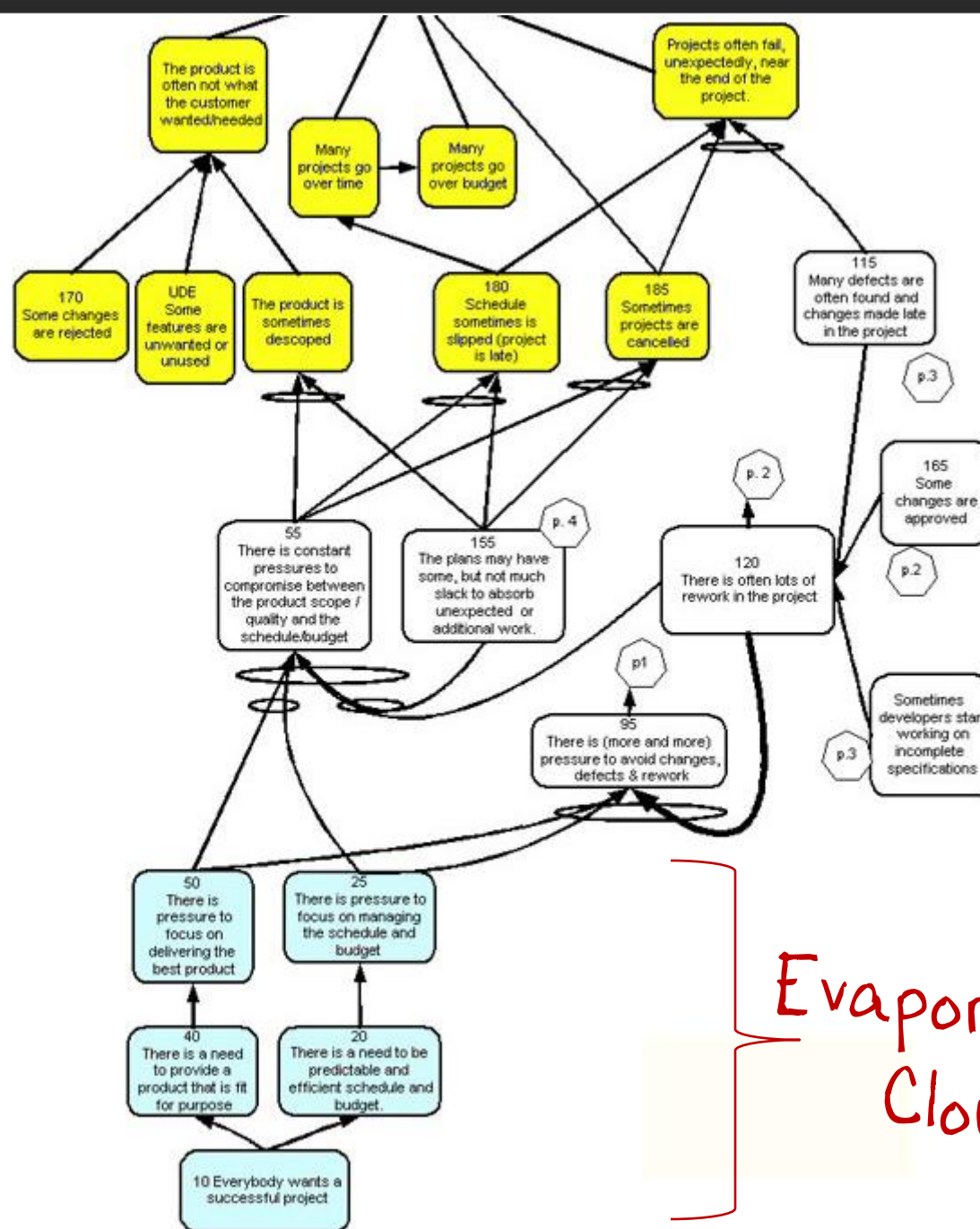
Fixing the W-Agilists



Enter Cindi, director of the company PMO

The Thinking Process





CRT

Evaporating Cloud



#1 - Lack of Meaningful Feedback Loops

- We are too busy to stop working!
- Daily stand-ups are useless, let's just use a single weekly status meeting.
- When we ask people for feedback they don't show up or don't participate anyway.
- We aren't really sure why we would want feedback or how we would use it anyway.



PRACTICE

Undesirable Effects

What are some of the undesirable effects the W-Agilists would see from the lack of feedback loops in their process?

1. Building wrong products
2. Building the product wrong
3. No improvement
4. Lack of visibility
5. Lack of trust
6. Key people out of the loop

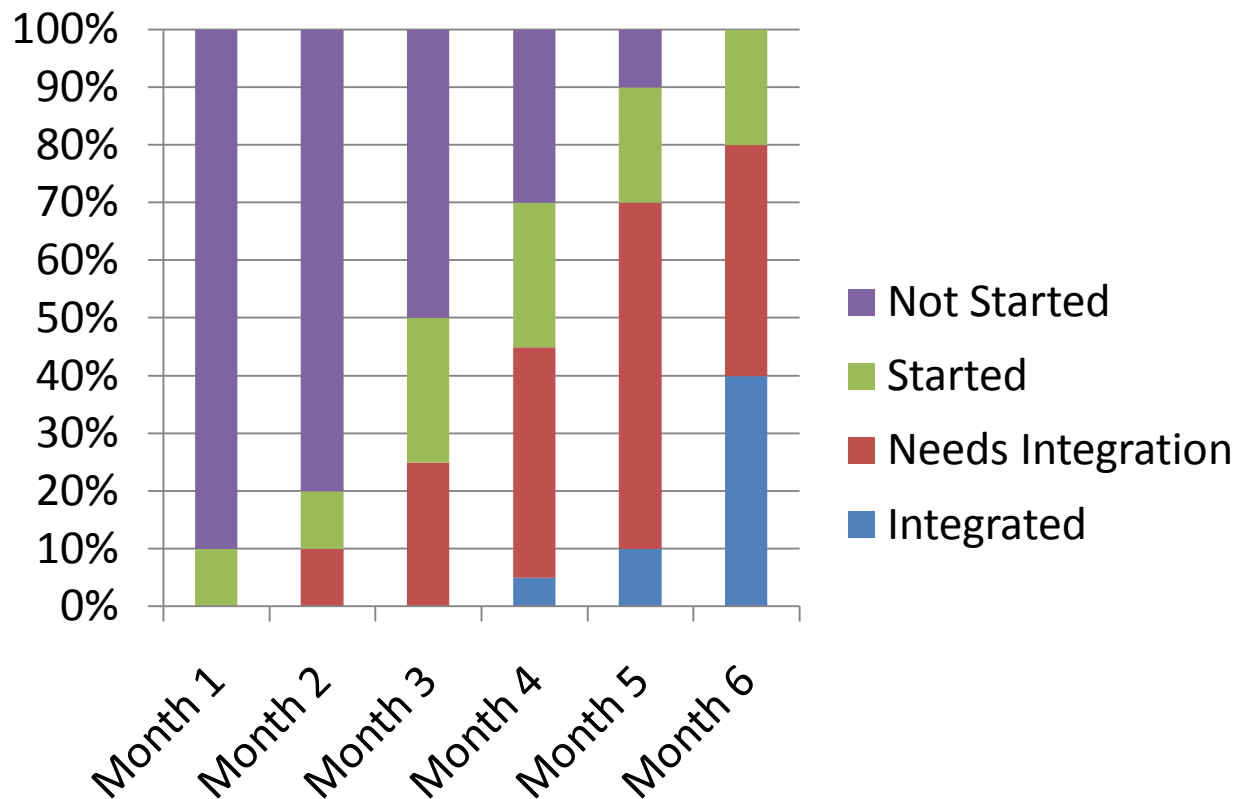
Phrase to remember

**Things can't be
improved without
regular feedback**

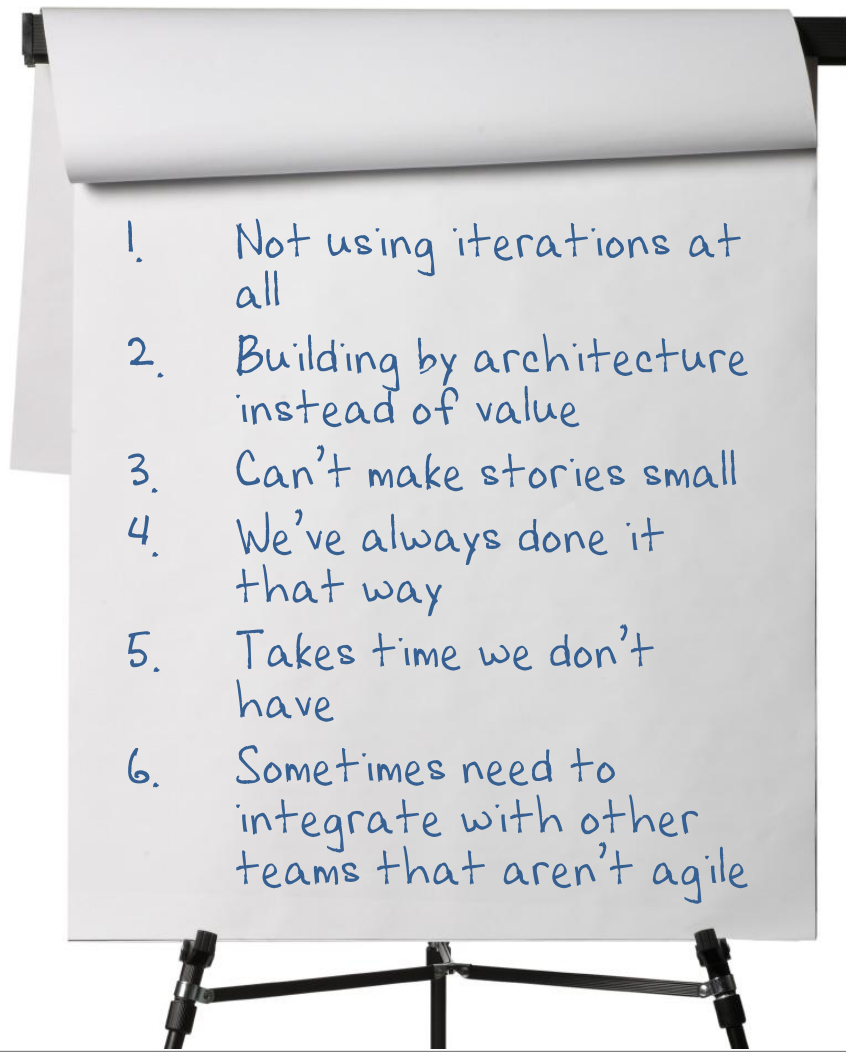


DEADLY SIN

#2 - No incremental deliveries of software



What might be some of the main causes of not building software iteratively (remember, this is supposed to be an agile team!)

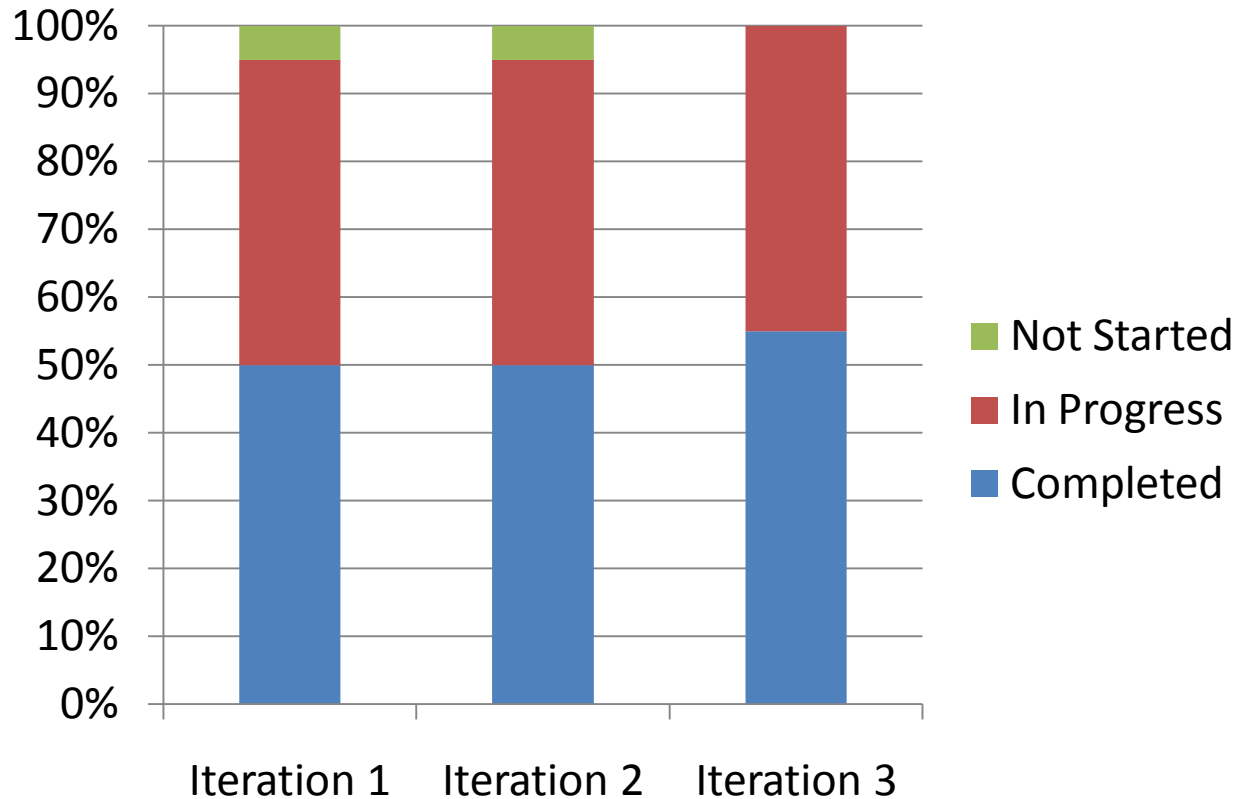
- 
1. Not using iterations at all
 2. Building by architecture instead of value
 3. Can't make stories small
 4. We've always done it that way
 5. Takes time we don't have
 6. Sometimes need to integrate with other teams that aren't agile

Phrase to remember

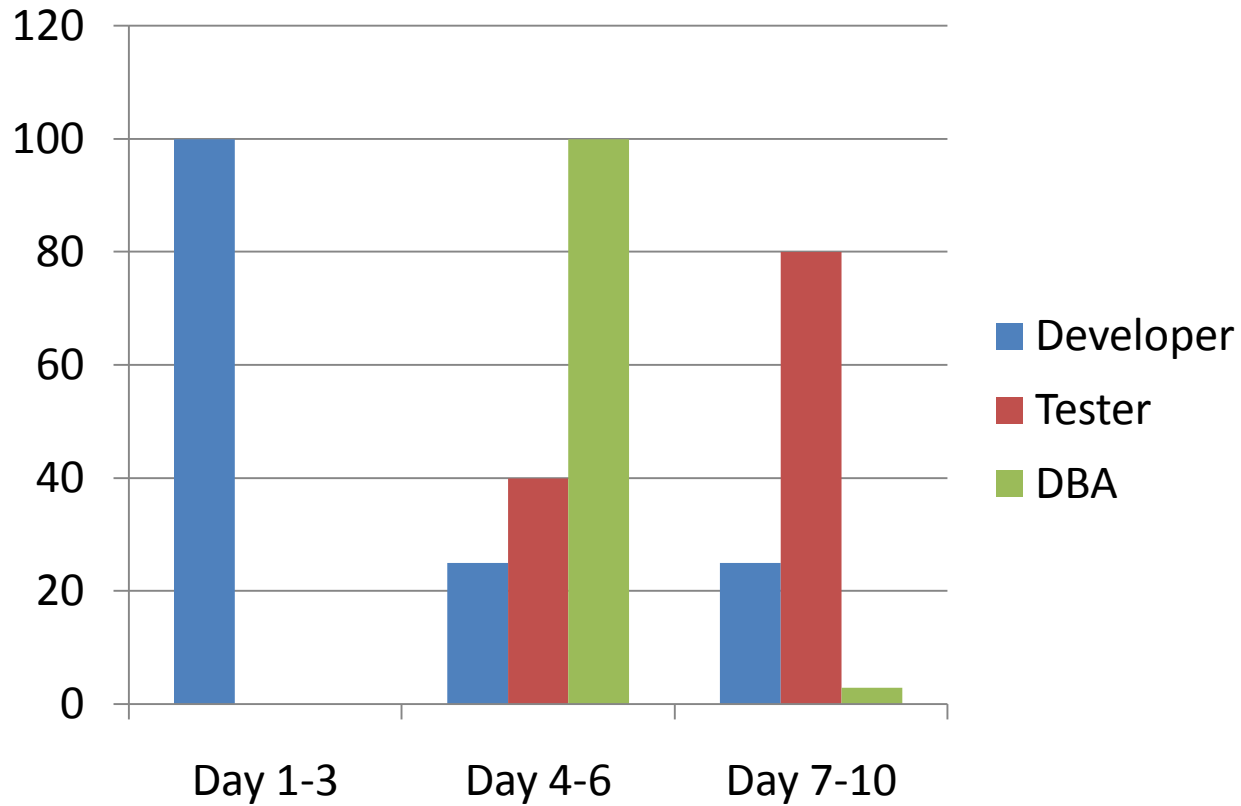
**Late integration will
lead to late shipping.**

**Instead deliver
completed value
every iteration.**

Our story continues...



Looking deeper...

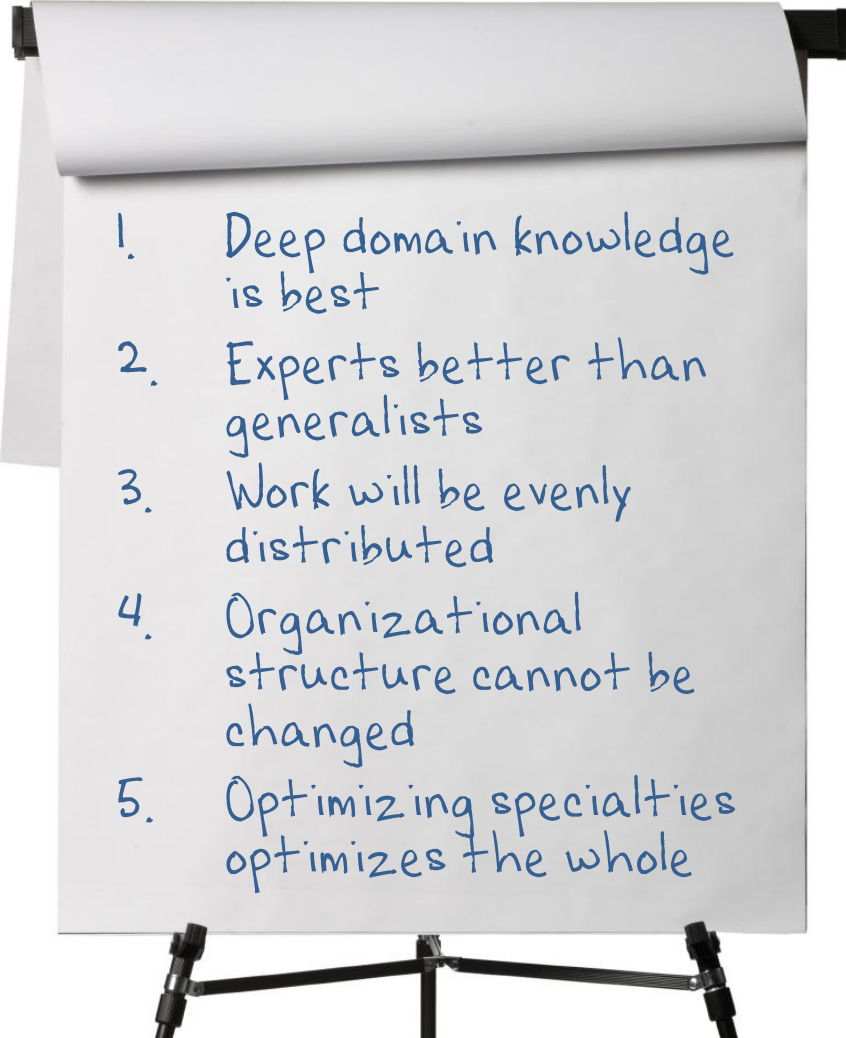




DEADLY SIN

What assumptions
cause many
organizations to
deliberately use silo'd
teams?

#3 - Silo'd Teams

- 
1. Deep domain knowledge is best
 2. Experts better than generalists
 3. Work will be evenly distributed
 4. Organizational structure cannot be changed
 5. Optimizing specialties optimizes the whole

Phrase to remember

**We don't care what
your business card
says, we care what
you can do for the
success of our team!**

The Evaporating Cloud

A tool for articulating and
breaking conflicts.





sticky~



resistance
to change?



malice?
incompetence?



Do X.

Don't do X.



Spend our
money on Y.

Spend our
money on Z.

Do X.

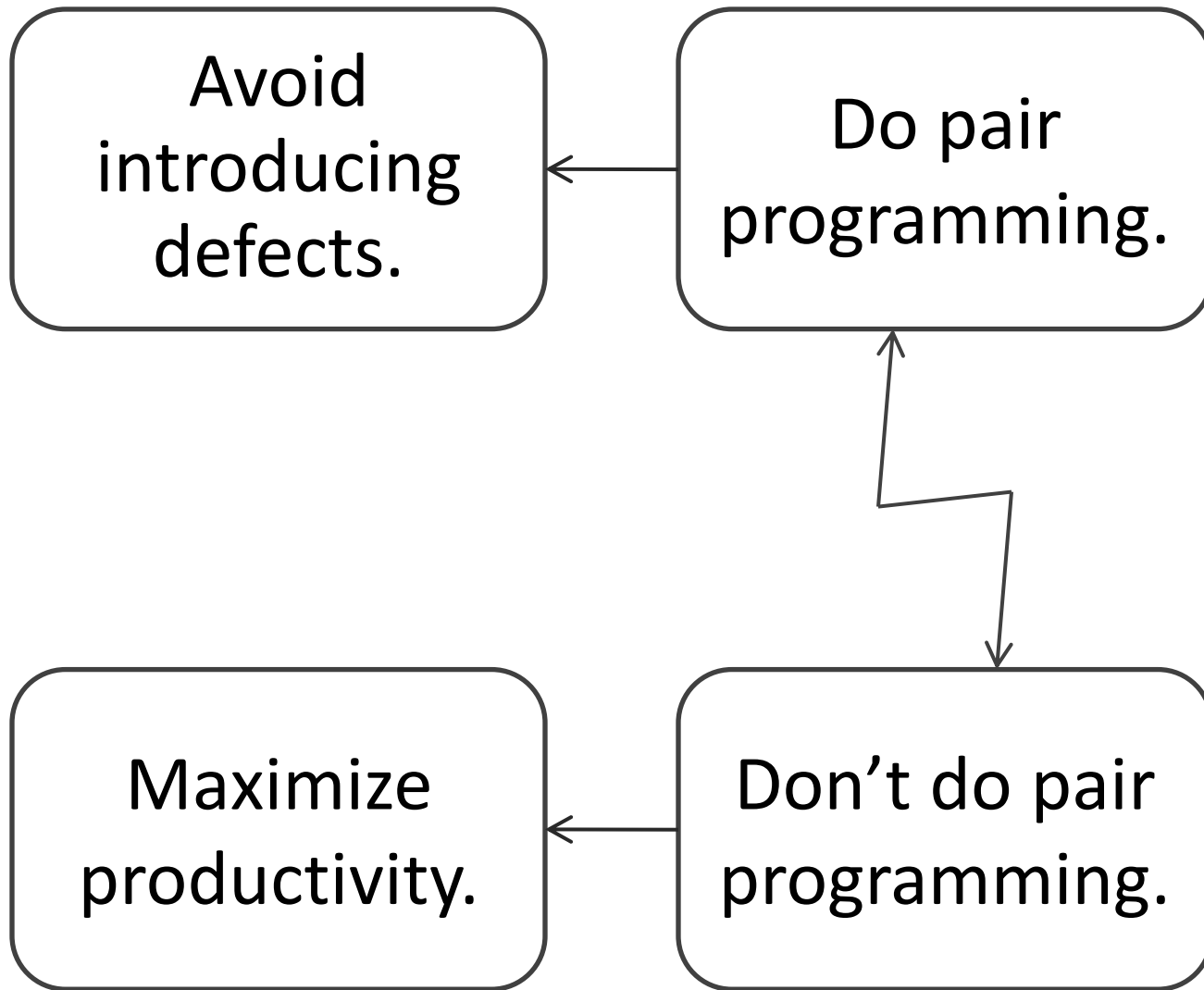
Don't do X.

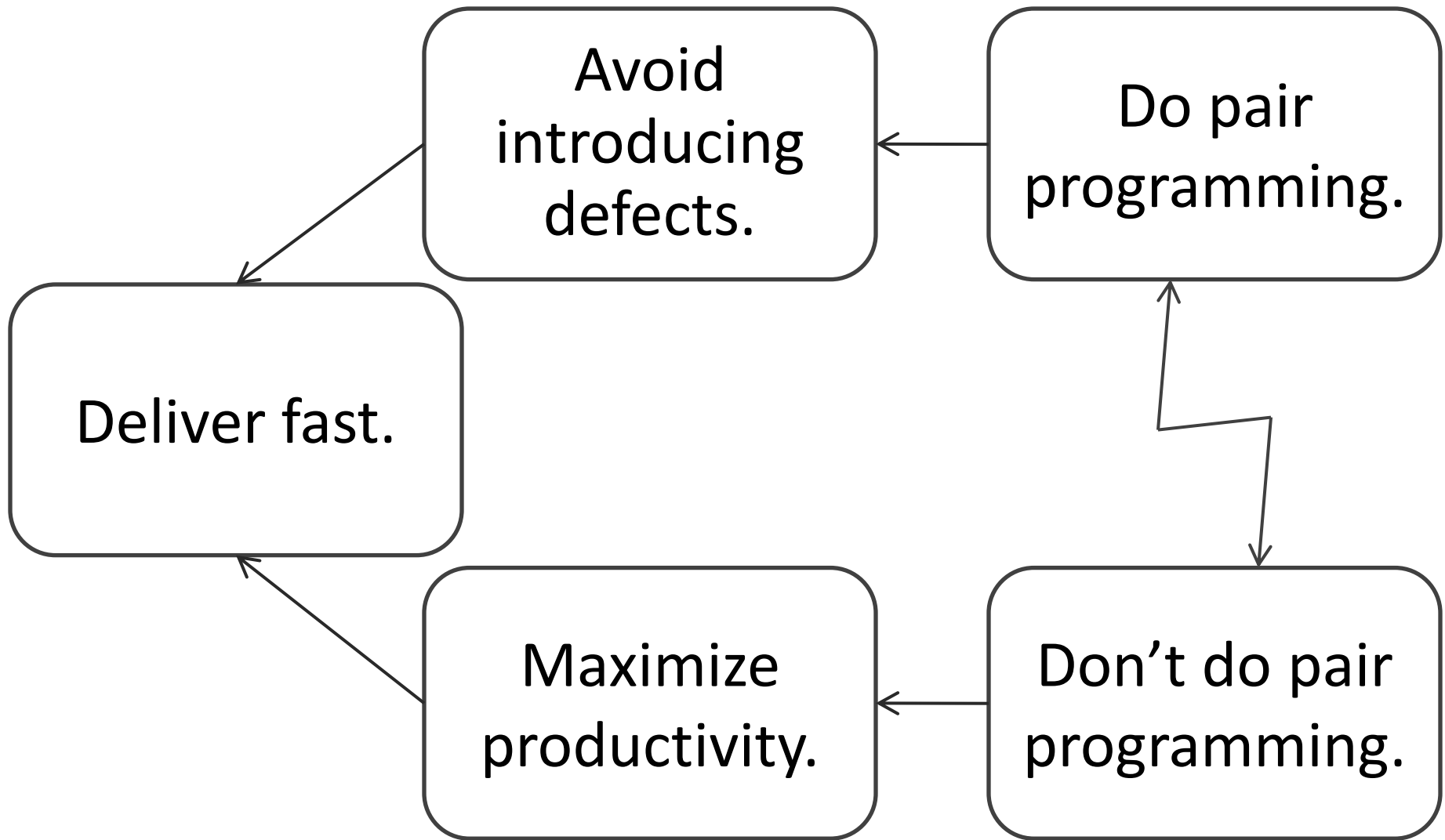


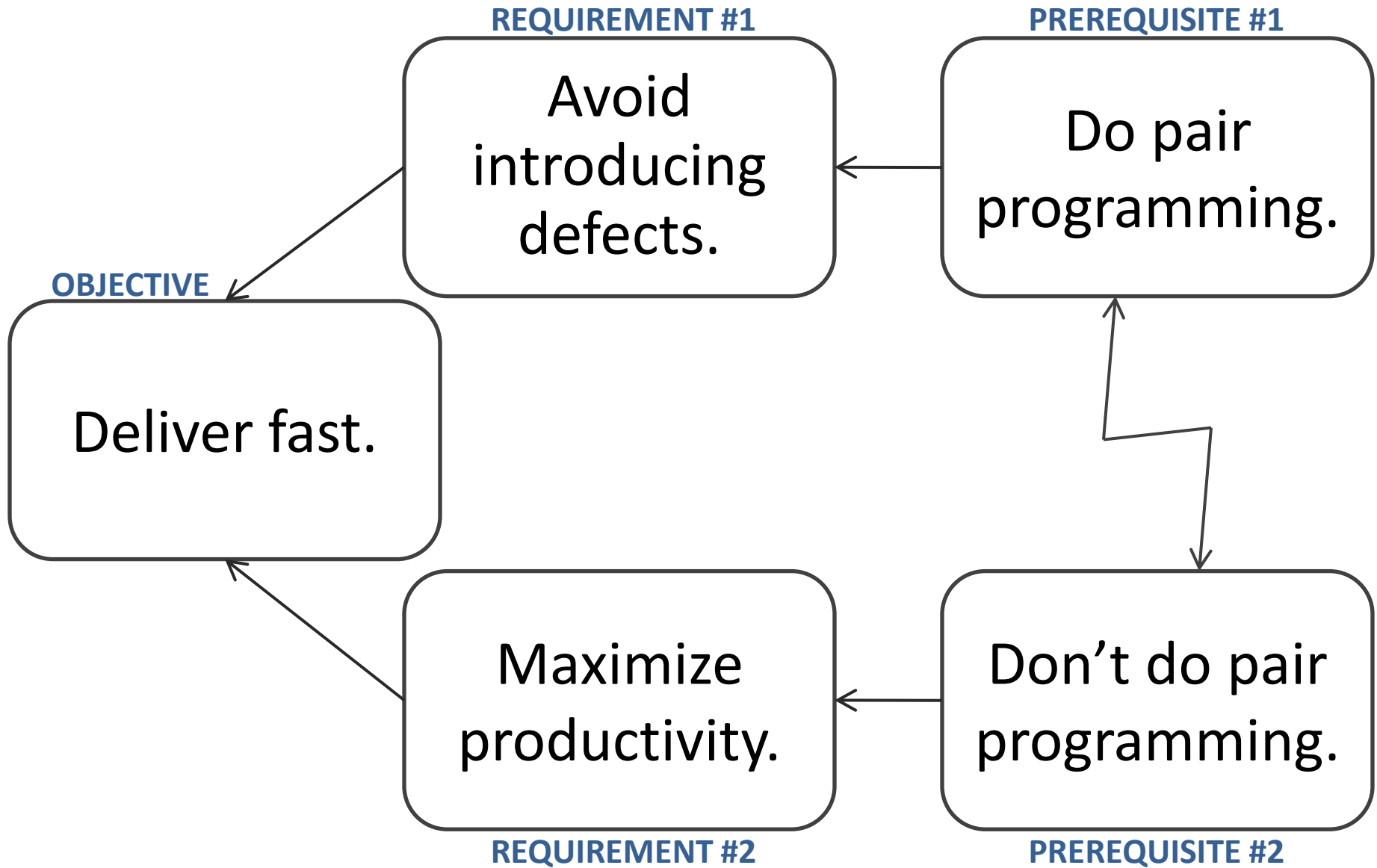
Do pair
programming.



Don't do pair
programming.

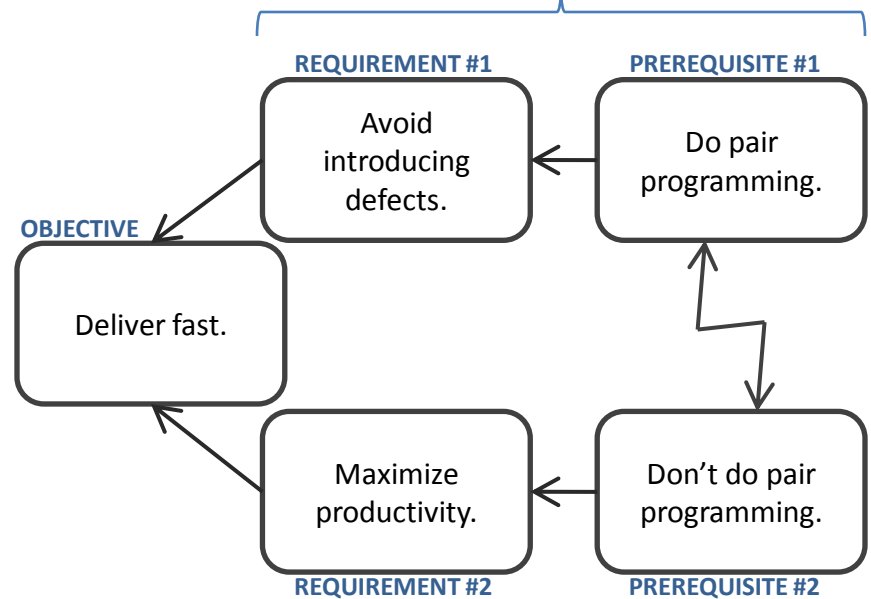






P1→R1: In order to avoid introducing defects,
we must do pair programming because...

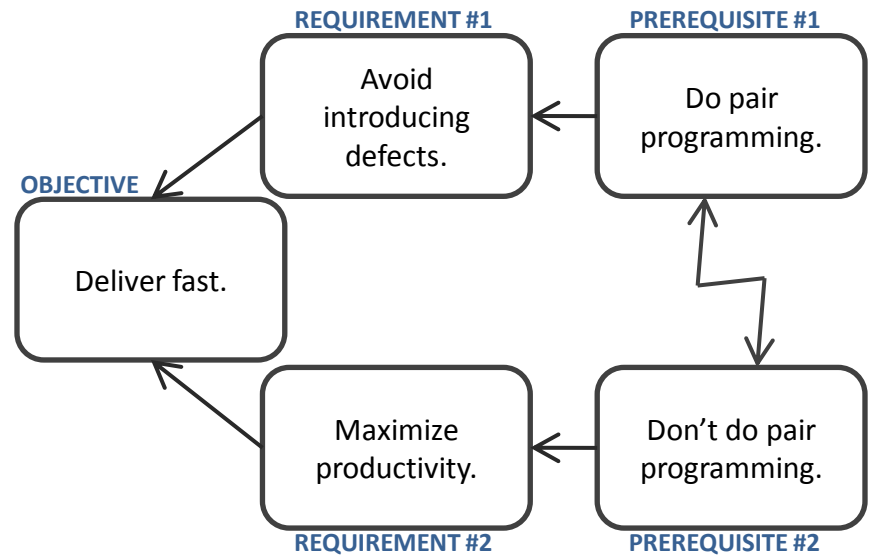
-
-
-
-
-



P1→R1: In order to avoid introducing defects, we must do pair programming because...

- Pair programming helps avoid introducing defects.

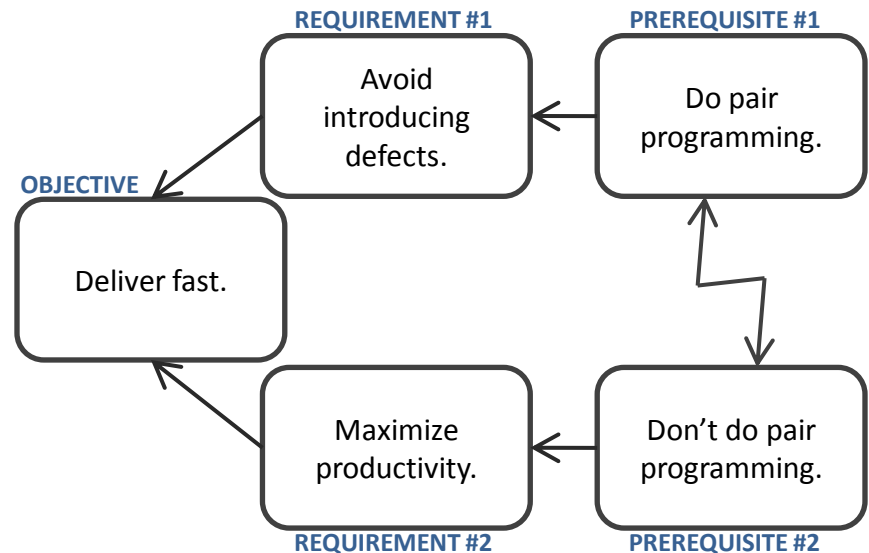
-
-
-
-



P1→R1: In order to avoid introducing defects, we must do pair programming because...

- Pair programming ~~helps~~ *is the ONLY way to* avoid introducing defects.

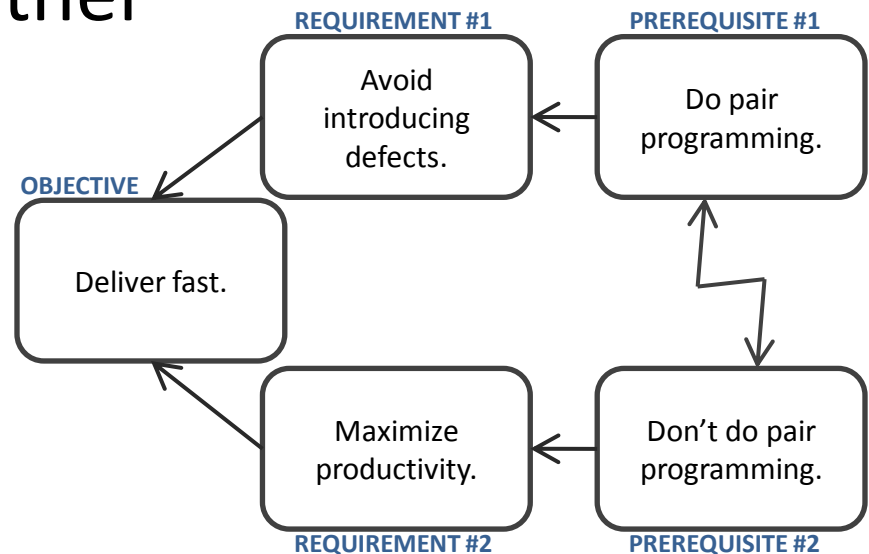
-
-
-
-



P1→R1: In order to avoid introducing defects, we must do pair programming because...

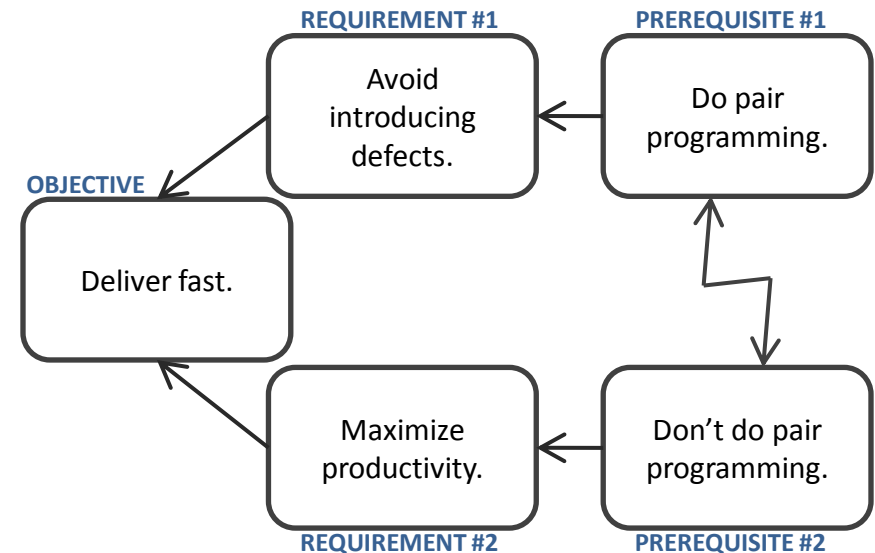
- Pair programming ~~helps~~ *is the ONLY way to* avoid introducing defects.
- Pair programming is better at avoiding defects than any other technique we know.
- Pair programming does other good things for us.

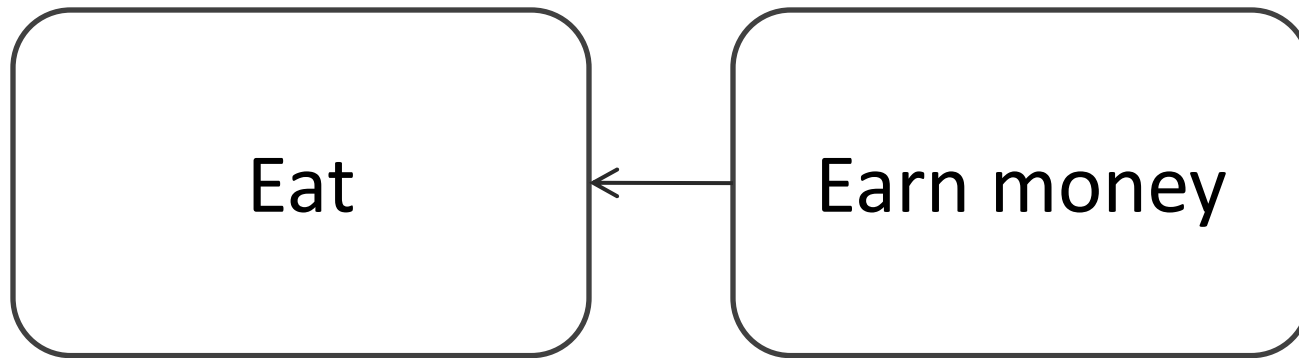
-
-



P2→R2: In order to maximize productivity, we must not do pair programming because...

- Pair programming ALWAYS reduces productivity.
- etc.
-
-
-



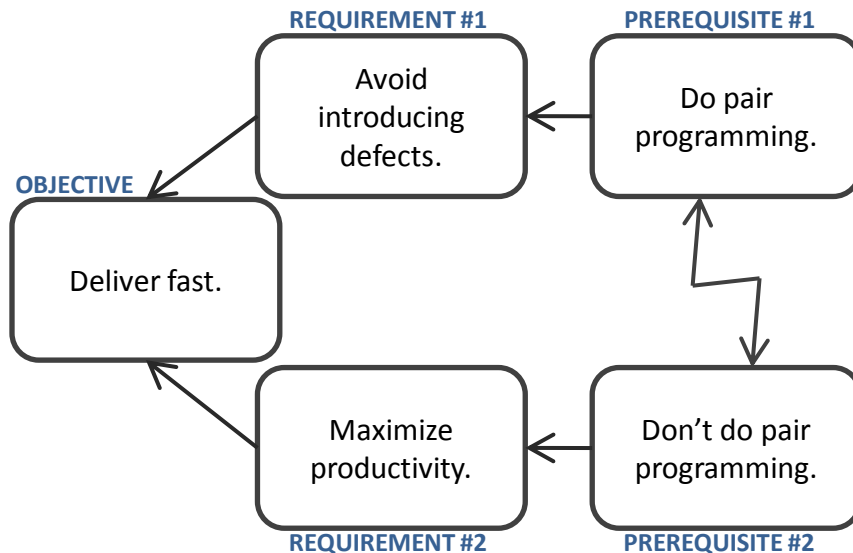


Assumptions:

- Buying food with money is the ONLY way to eat.
- The ONLY way to get money is to earn it.

How else could we eat without earning money?

How else could we avoid introducing defects without doing pair programming?



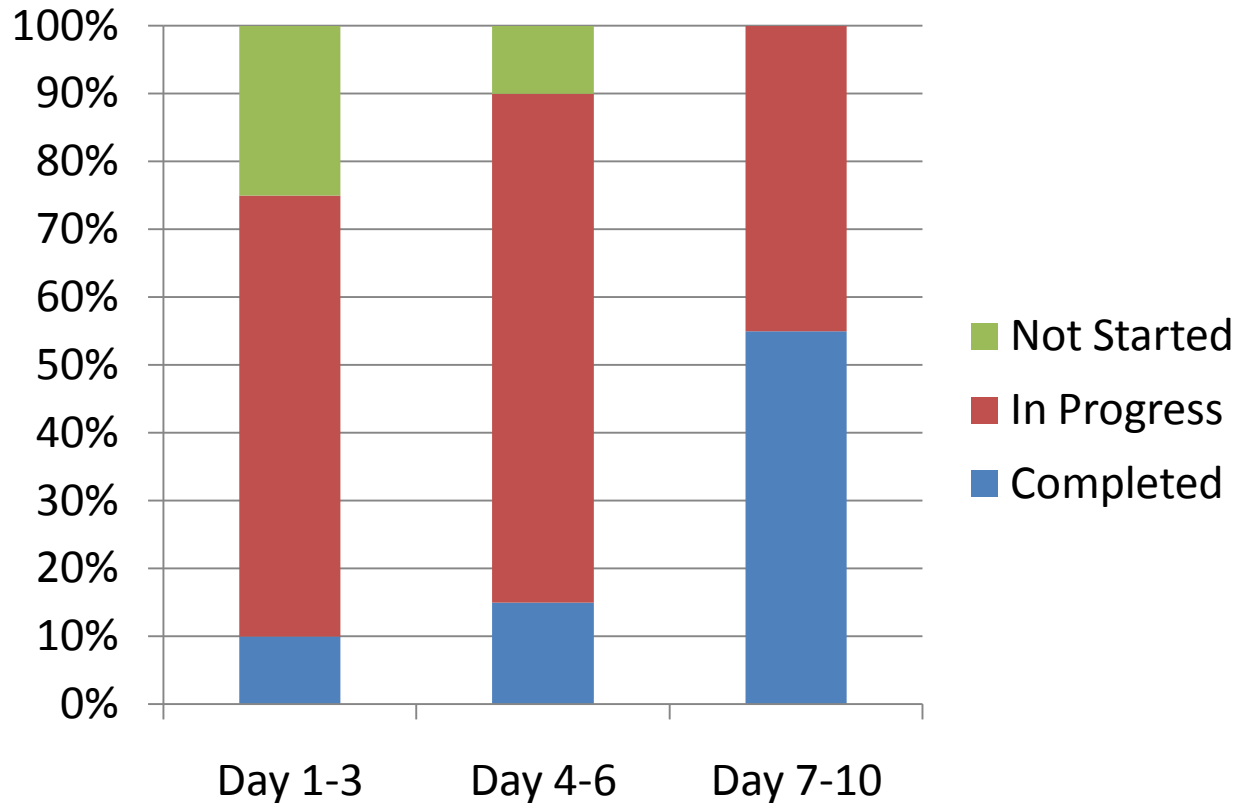
How could we do *and* not do pair programming?

How else could we maximize productivity *and* do pair programming?

Back to Our Fictional Team

W-Agilists

Digging deeper in an iteration...

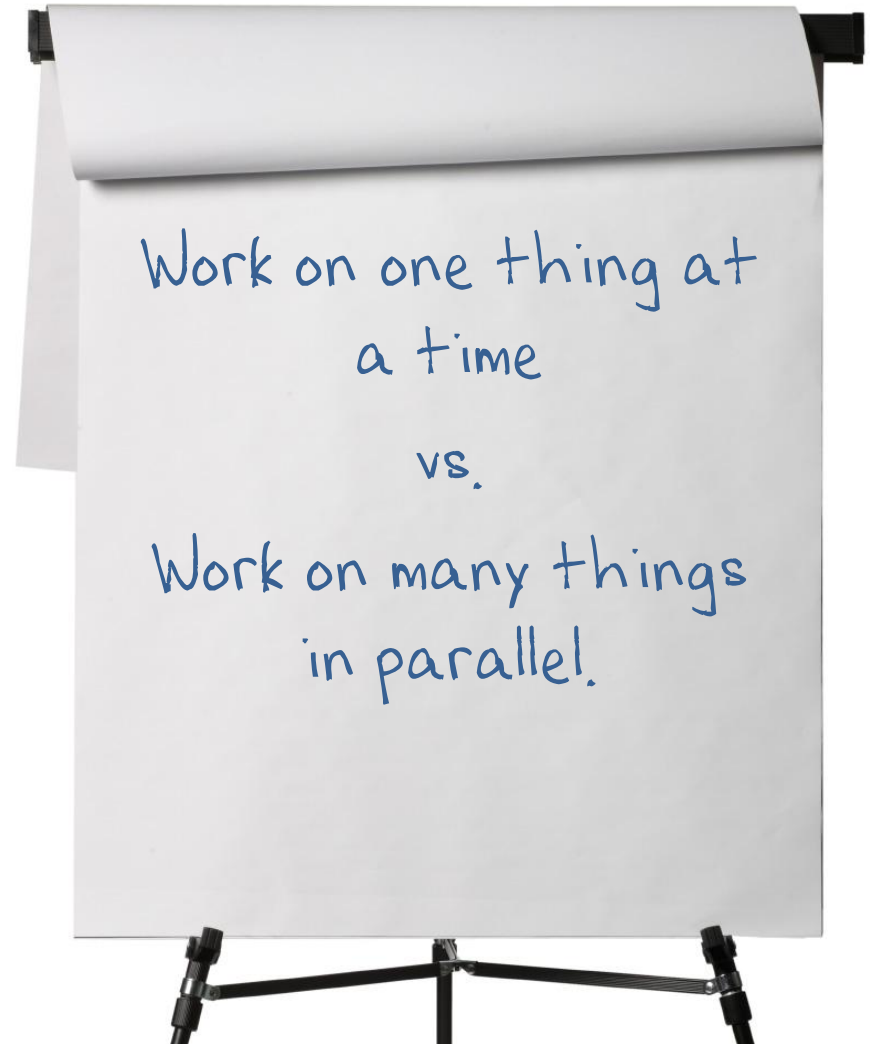


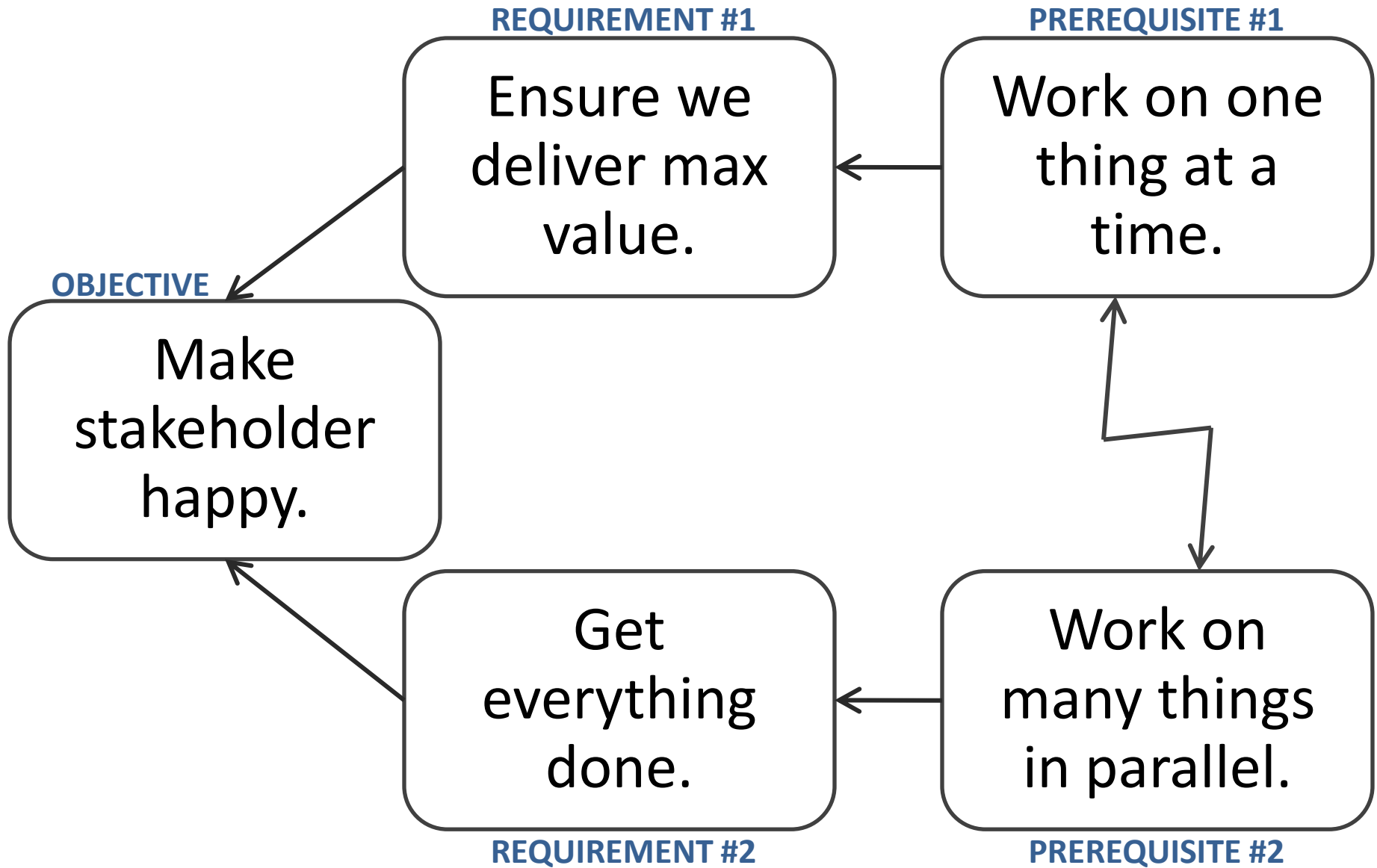


DEADLY SIN

#4 - Too much work in process

Using the evaporating cloud from the Thinking Process, let's solve this conflict together.





Phrase to remember

WIP = Waste in Progress!

Presenting an Evaporating Cloud



First, build the whole cloud

Assumptions:

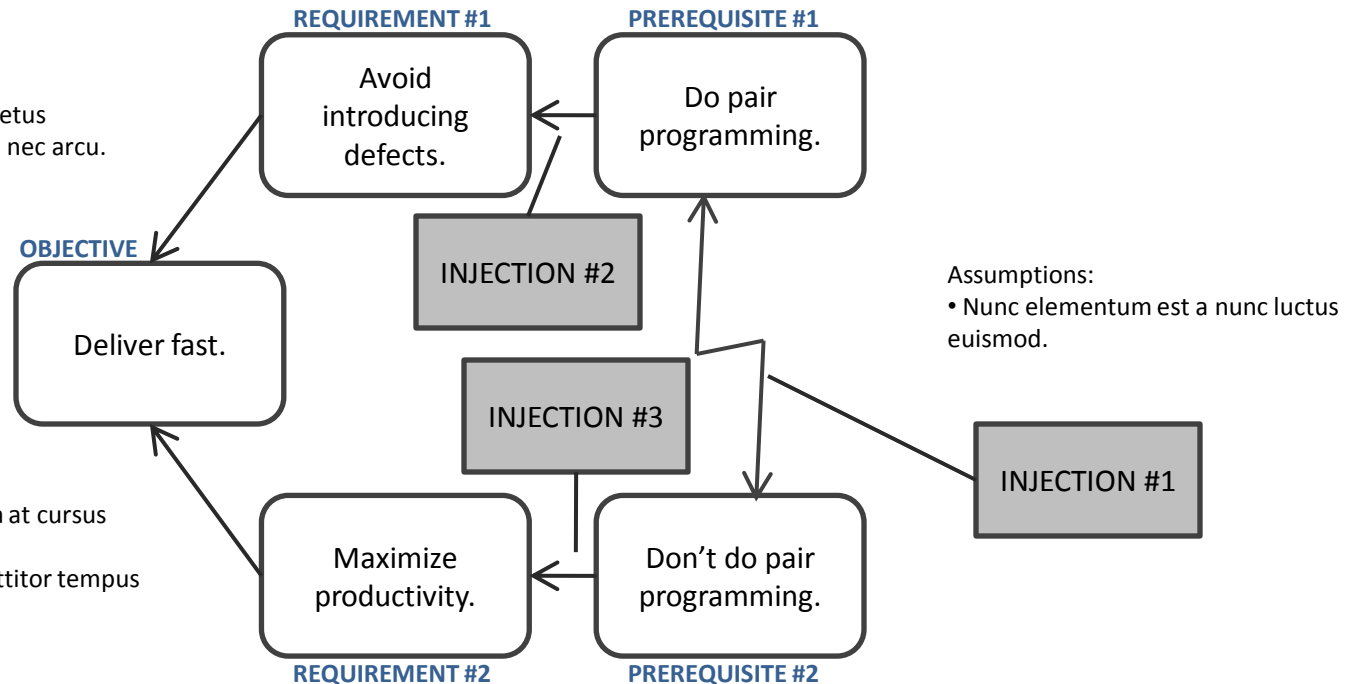
- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Integer id libero vitae dolor ornare condimentum pellentesque ut mauris.
- Duis iaculis metus ut arcu tempor lobortis.
- Integer interdum pellentesque orci, sit amet sollicitudin lectus vehicula nec.

Assumptions:

- Aliquam vitae sem id metus imperdiet commodo non nec arcu.

Assumptions:

- Duis pulvinar orci at eros pharetra at cursus nulla ultrices.
- Pellentesque et eros at quam porttitor tempus id et leo.



Assumptions:

- Aenean vehicula lacinia lacus, sed hendrerit erat imperdiet quis.
- Fusce at tortor in orci convallis porttitor sit amet non magna.
- Aenean sit amet lacus nec justo consequat auctor.

Then, redo the assumptions and injections together

Assumptions:

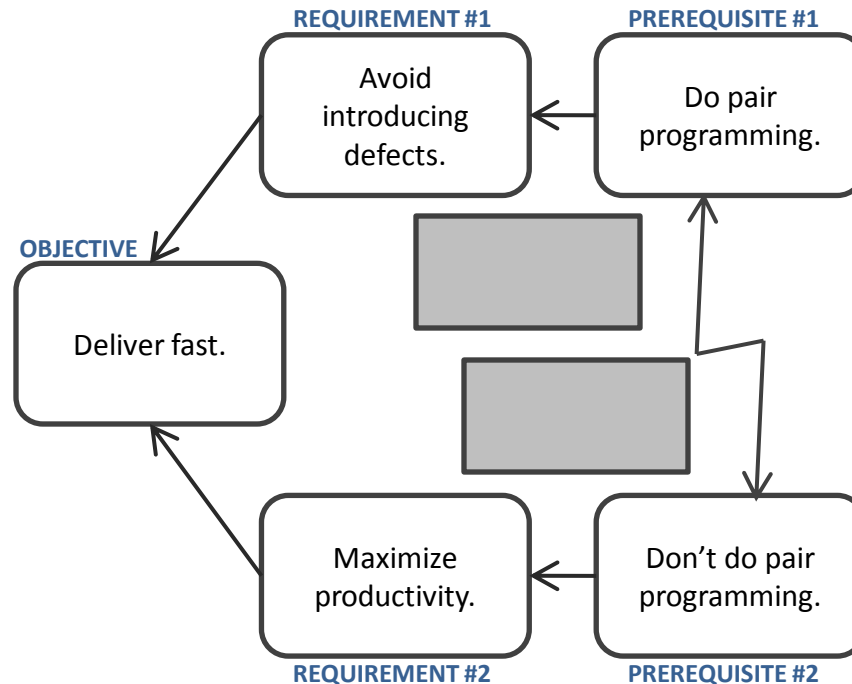
-
-
-
-

Assumptions:

-
-
-

Assumptions:

-
-
-



Assumptions:

-
-

Assumptions:

-
-
-
-

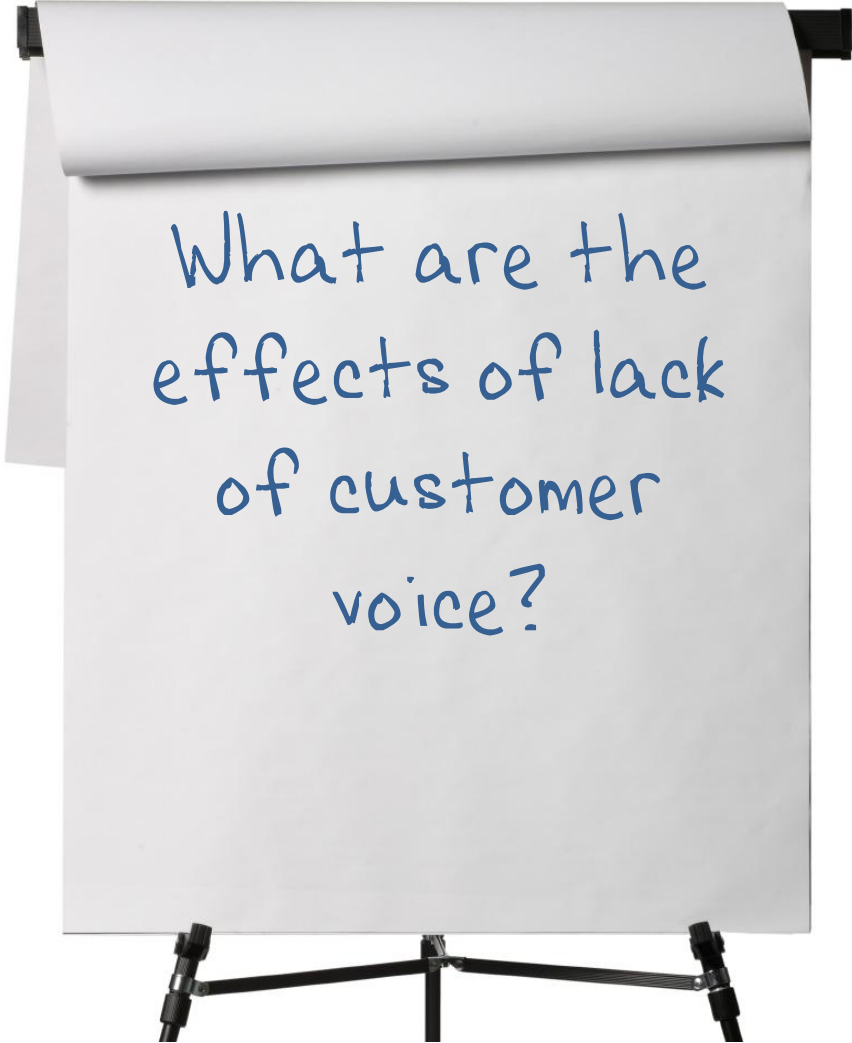


DEADLY SIN

#5 - Lack of customer voice



Discuss



What are the
effects of lack
of customer
voice?



PRACTICE

Lack of customer voice

Create an evaporating cloud in your group

Engage the customer

vs.

Don't engage the customer

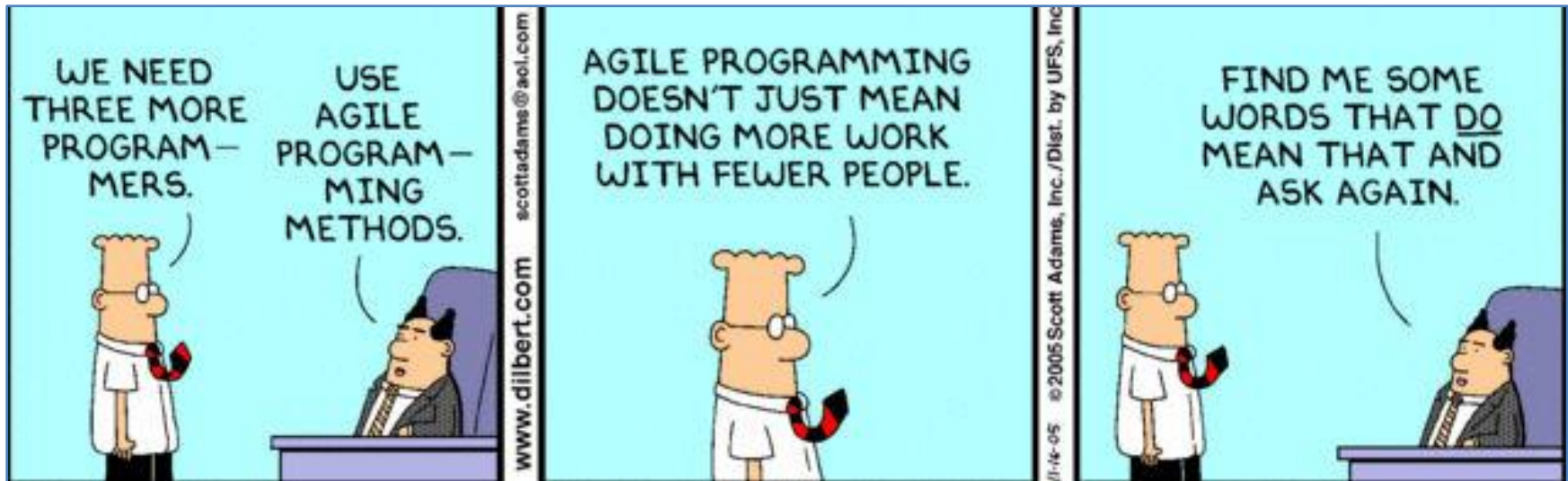
Phrase to remember

**Build the simplest
thing that works –
then get real
customer feedback!**

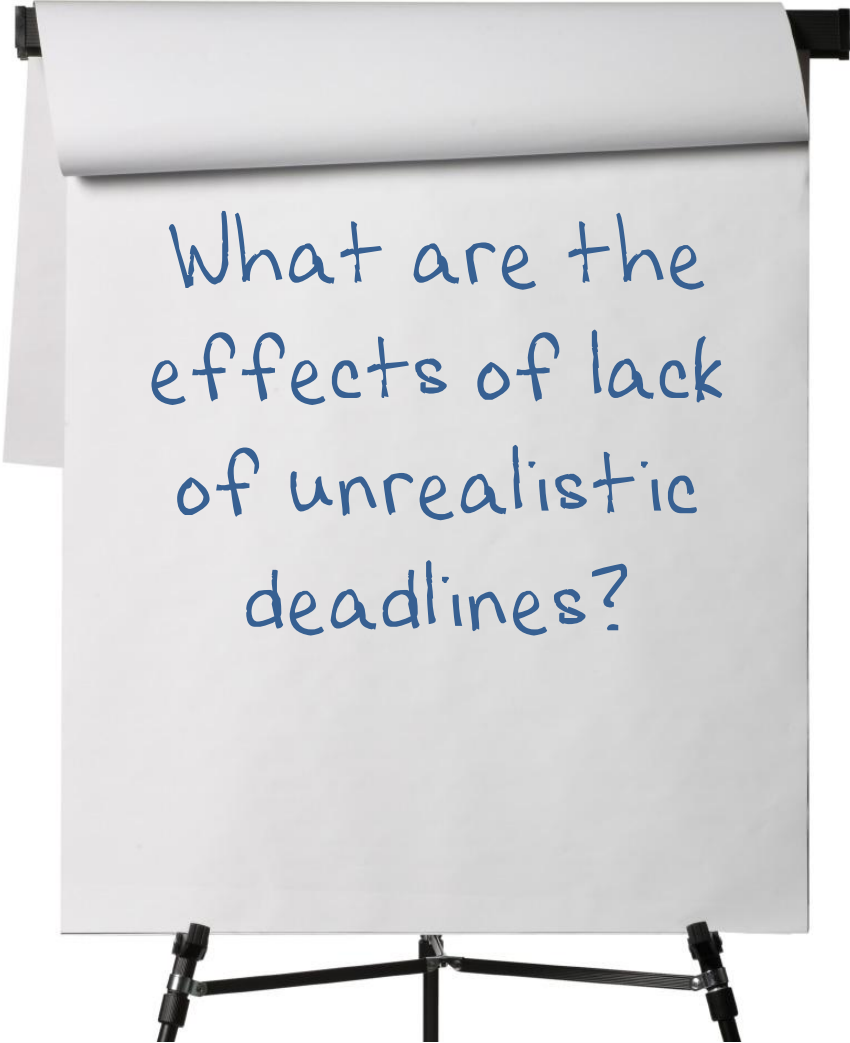


DEADLY SIN

#6 - Unrealistic deadlines



Discuss

A flipchart on a stand with a handwritten question.

What are the
effects of lack
of unrealistic
deadlines?

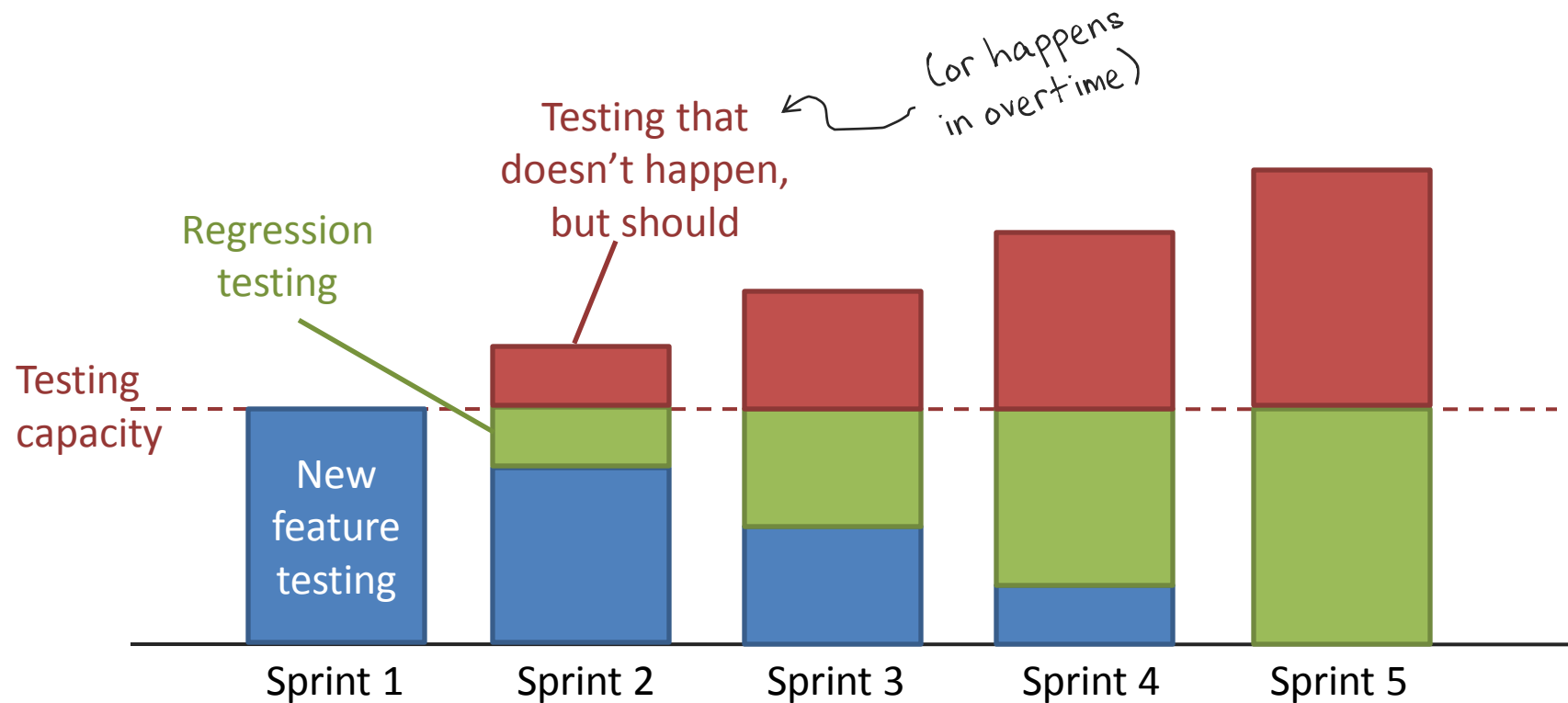
Phrase to remember

**Customer/stakeholder
needs do not alter the
realities of the universe.**

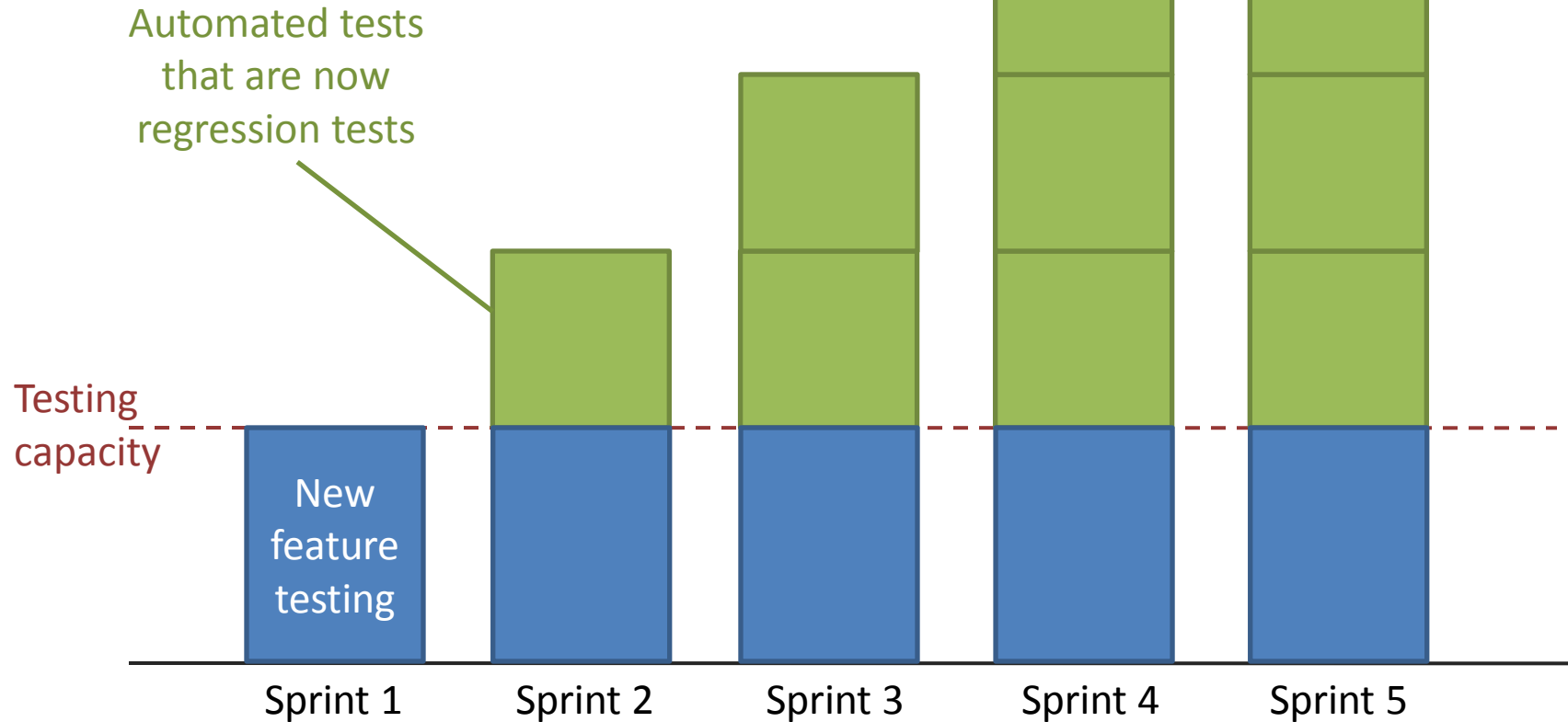


DEADLY SIN

#7 - Manual testing taking too long



Automate and get regression tests for (nearly) free



Phrase to remember

**Automate any tests that
will run more than once.**

**Do you know ahead of
time which tests those
are???**



PRACTICE

Unrealistic Deadlines OR Adopting Automated Testing

Create an evaporating
cloud in your group.

Adopt automated
testing vs. Don't
adopt...

or

Say "yes" to all
customer demands vs.
Say "no" to some...

Recap of the 7 Deadly Sins

1. Missing feedback loops
2. Not building in iterations – large scale integrations
3. Silo'd teams
4. Too much work-in-progress (WIP)
5. Lack of customer voice
6. Unrealistic deadlines
7. Over-reliance on manual testing

Questions?

www.agilebob.com

bob.hartman@agileforall.com

303-766-0917

www.richardlawrence.info

richard@humanizingwork.com

303-895-7688