

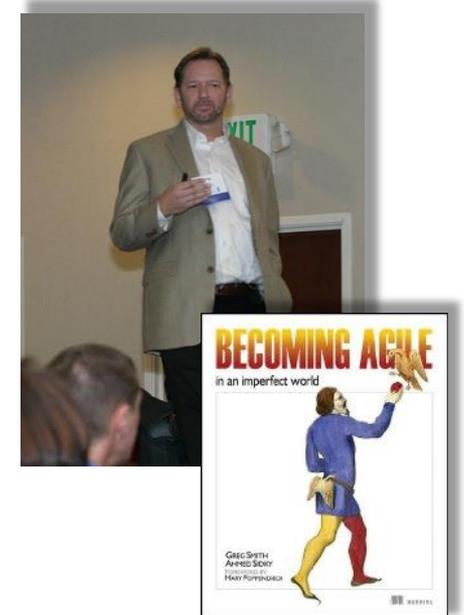
# The Top Seven Reasons Sprints Fail

....and What To Do About It



# Agile Coach/Trainer – Greg Smith

- Founder/Agile Coach - GS Solutions Group
- Helped numerous government agencies move to Agile including Fannie Mae and New York City IT
- Helped over 100 companies move to Agile including Exxon, Microsoft, Chase, TD Ameritrade, Halliburton, StubHub, and Expeditors International.
- Author – ***Becoming Agile in an Imperfect World***  
(part of the curriculum for PMI Agile certification)
- Experienced in Agile, Lean and Traditional methods



[greg@gssolutionsgroup.com](mailto:greg@gssolutionsgroup.com)

AgileSmith



# In Olympia on May 21st

## 2014 IPMA Forum

### Event Date and Time:

May 20 2014 - 7:30am - May 21 2014 - 4:30pm

[Register Now!](#)

[Download Forum Flyer](#)

## Welcome to Forum 2014

**"Business and Technology Working Together to Solve Problems"**

**May 20-21, 2014 at Saint Martin's University in Lacey WA.**

## Assimilating Agile, Lean, and Traditional Project Management

### Track:

Professional Development

### Date/Time/Room:

Wednesday May 21st / 1:00 pm - 2:00 pm / Conrad Room

### Target audience:

IT managers, program managers, project managers, business analysts, and business SMEs involved in IT technology or lean improvement projects.

### Speaker(s):

Todd Williams, PMP, CEO, eCameron, inc., Panel Moderator

Greg Smith, PMP-ACP, CEO, GS Solutions Group, Agile Panelist

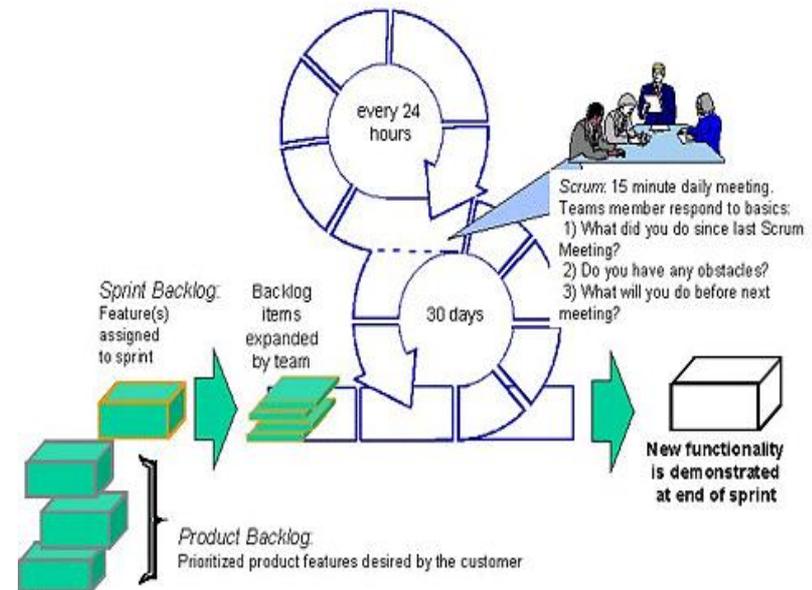
Jim Benson, PMP, CEO, Modus Cooperandi, Lean Panelist

Samad Aidane, PMP, Managing Director, NeuroFrontier.com, PMP Panelist

Jack Tollefson, President, TeamStronger, Workshop Producer/Organizer/Announcer

# What Are Sprints?

- A time-boxed event that delivers a subset of *production ready* software
- The software could be custom code, or a configured *off the shelf* product such as SAP, Siebel, or Sharepoint
- A team is usually dedicated to a sprint. They all work together to deliver the software by the end of the sprint.
- A customer representative, called a *product owner*, interacts with the team during the sprint to answer questions and to clarify requirements as needed.
- *Note that the sprint process can be used for non-software projects and often is.*



*Sprints are typically 1 to 4 weeks, but I have seen them as long as 8 weeks and still being beneficial.*

# Why Sprints?

- We can pursue delivering value to the customer throughout a project, as opposed to just the end
- We do a better job of delivering accurately due to a shorter gap between analysis, coding, and testing
- We have a logical place to adapt to discoveries throughout a project (at the end of a sprint)
- We have a great tool for measuring the most important project metric – *how much code is truly complete?* (transparency)



# What Is Sprint Failure?

- Missing the date
- Inconsistent understanding
- Delivering the wrong thing
- Exhausting the team
- Losing focus
- Working as departments
- Not learning
- Lack of discipline
- Excessive waste or delays



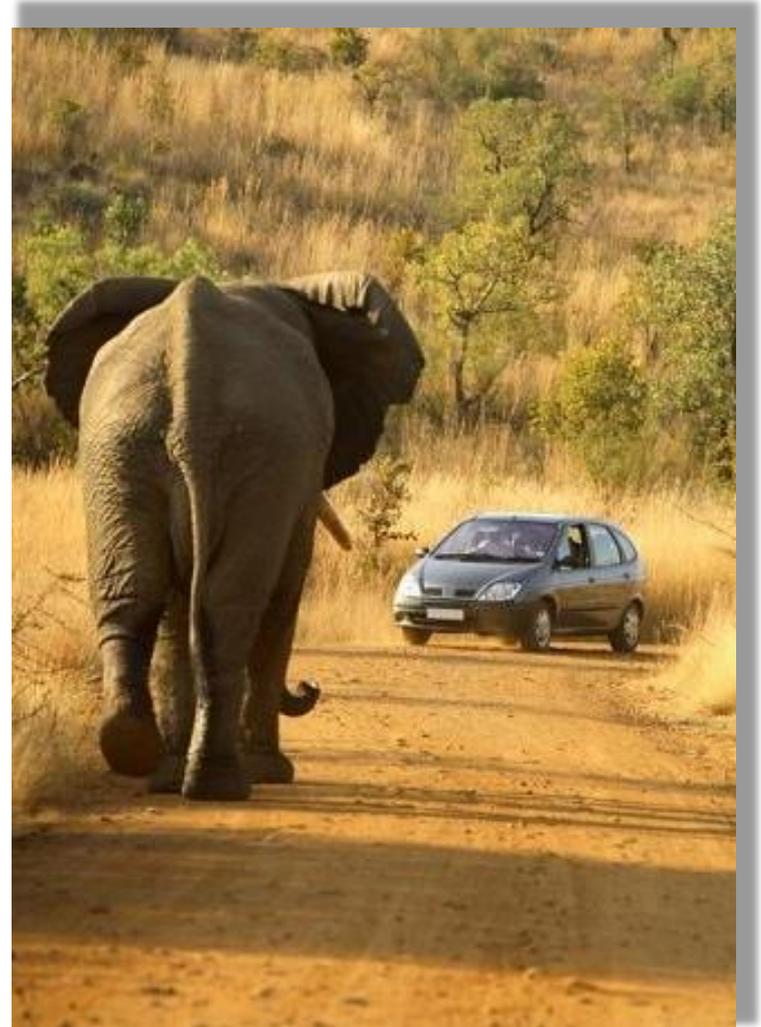
# Education Failure

- Teams Often Fail Due to Lack of Knowledge
- A lot of folks using Agile based on a book
- There are numerous “i”s to dot and “t”s to cross
- I hope to *dot* and *cross* some with you today



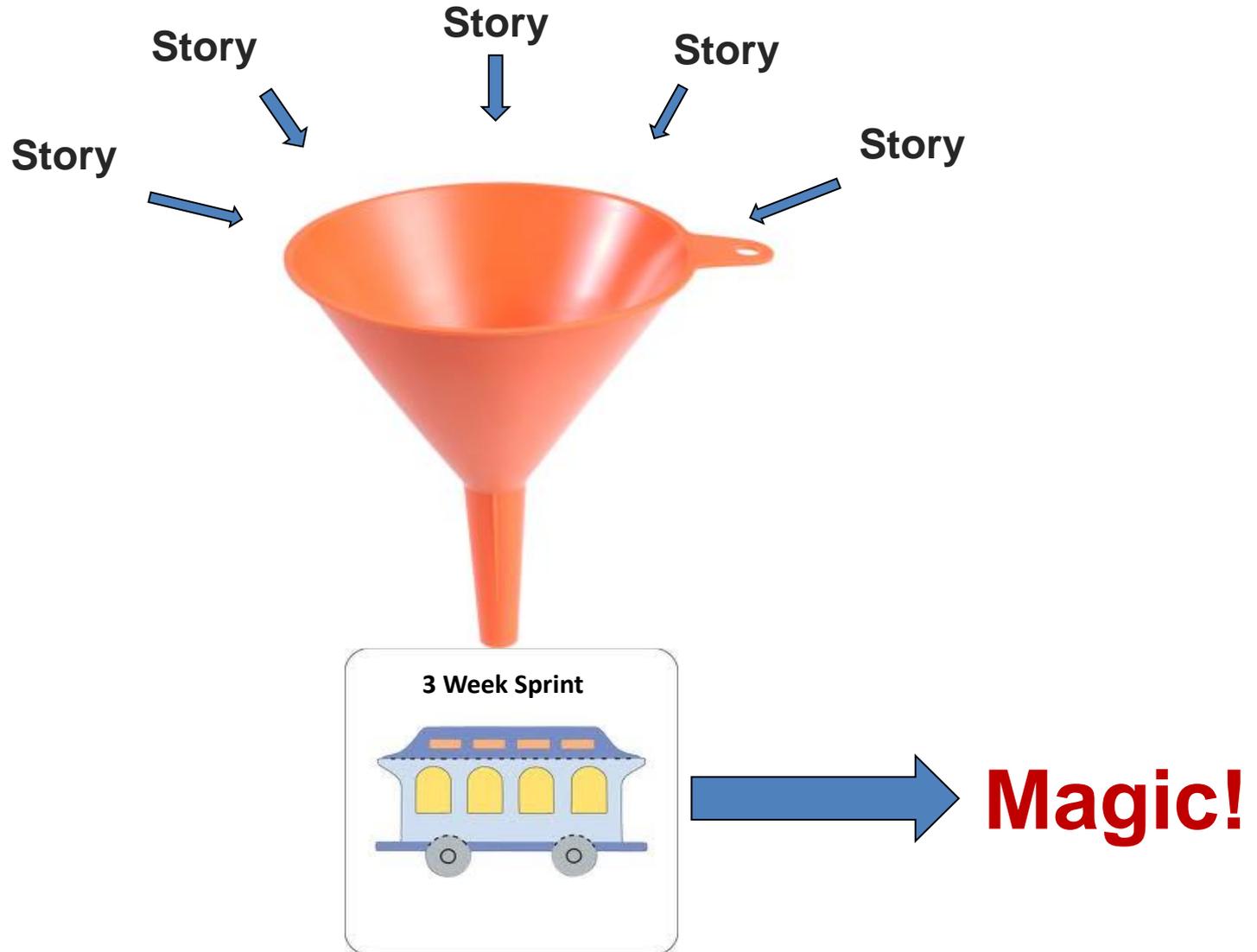
# Situation Failure

- Many teams have constraints that impact sprints
- Offshore resources
- Non-dedicated resources
- Cameo customers
- Governance
- Regulations
- Volatile business environment
- Green team



***There is usually a way to subvert a situation or minimize the impact to the sprint***

# Issue 1 – No Sprint Structure

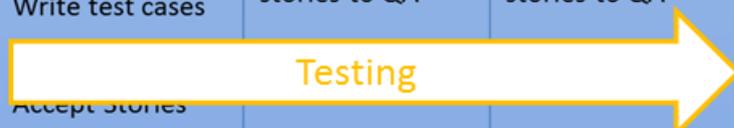
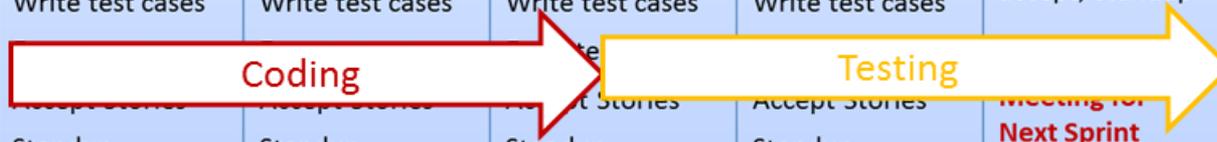


# Structure Within a Sprint

	Day 1	Day 2	Day 3	Day 4	Day 5
<b>Week 1:</b> <b>Sprint Planning &amp; Start</b>	POs review stories with Dev & QA	<ul style="list-style-type: none"> <li>Devs and QA Estimate.</li> <li>Determine build sequence.</li> <li>Finalize Acceptance Criteria</li> <li>COMMIT!</li> </ul>	Build Write test cases Execute tests Accept Stories Standup	Build Write test cases Execute tests Accept Stories Standup	Build Write test cases Execute tests Accept Stories Standup
<b>Week 2:</b> <b>BUILD</b>  <b>TEST</b>  <b>ACCEPT</b>	Build Write test cases Execute tests Accept Stories Standup	Build Write test cases Execute tests Accept Stories Standup	Build Write test cases Execute tests Accept Stories Standup	Build Write test cases Execute tests Accept Stories Standup	Build, test, accept, standup  <b>Grooming Meeting for Next Sprint</b>
<b>Week 3</b> <b>Finalize</b>  <b>Demo</b>  <b>Learn</b>	Build Write test cases Execute tests Accept Stories Standup	Harden, no new stories to QA	Harden, no new stories to QA	Review with Stakeholders  Finalize Stories for Next Sprint  Review velocity	Retrospective  Update regression test list  Update release plan*

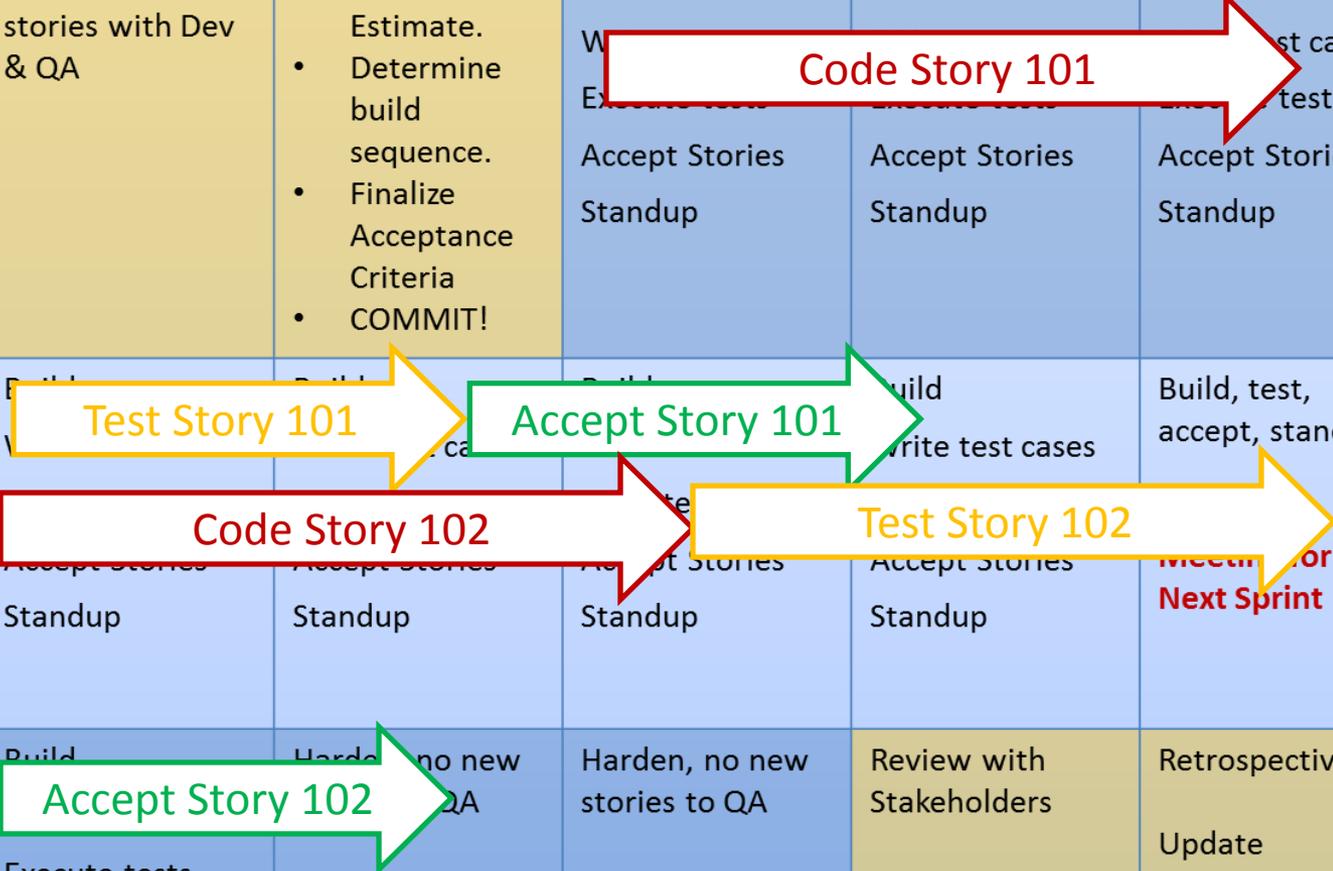
# Leads to Issue 2 – Waterfall Within a Sprint

	Day 1	Day 2	Day 3	Day 4	Day 5
<b>Week 1: Sprint Planning &amp; Start</b>	POs review stories with Dev & QA	<ul style="list-style-type: none"> <li>Devs and QA Estimate.</li> <li>Determine build sequence.</li> <li>Finalize Acceptance Criteria</li> <li>COMMIT!</li> </ul>	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup
<b>Week 2: BUILD  TEST  ACCEPT</b>	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build, test, accept, standup Meeting for Next Sprint
<b>Week 3 Finalize  Demo  Learn</b>	Build Write test cases Accept Stories Standup	Harden, no new stories to QA	Harden, no new stories to QA	Review with Stakeholders Finalize Stories for Next Sprint Review velocity	Retrospective Update regression test list Update release plan*



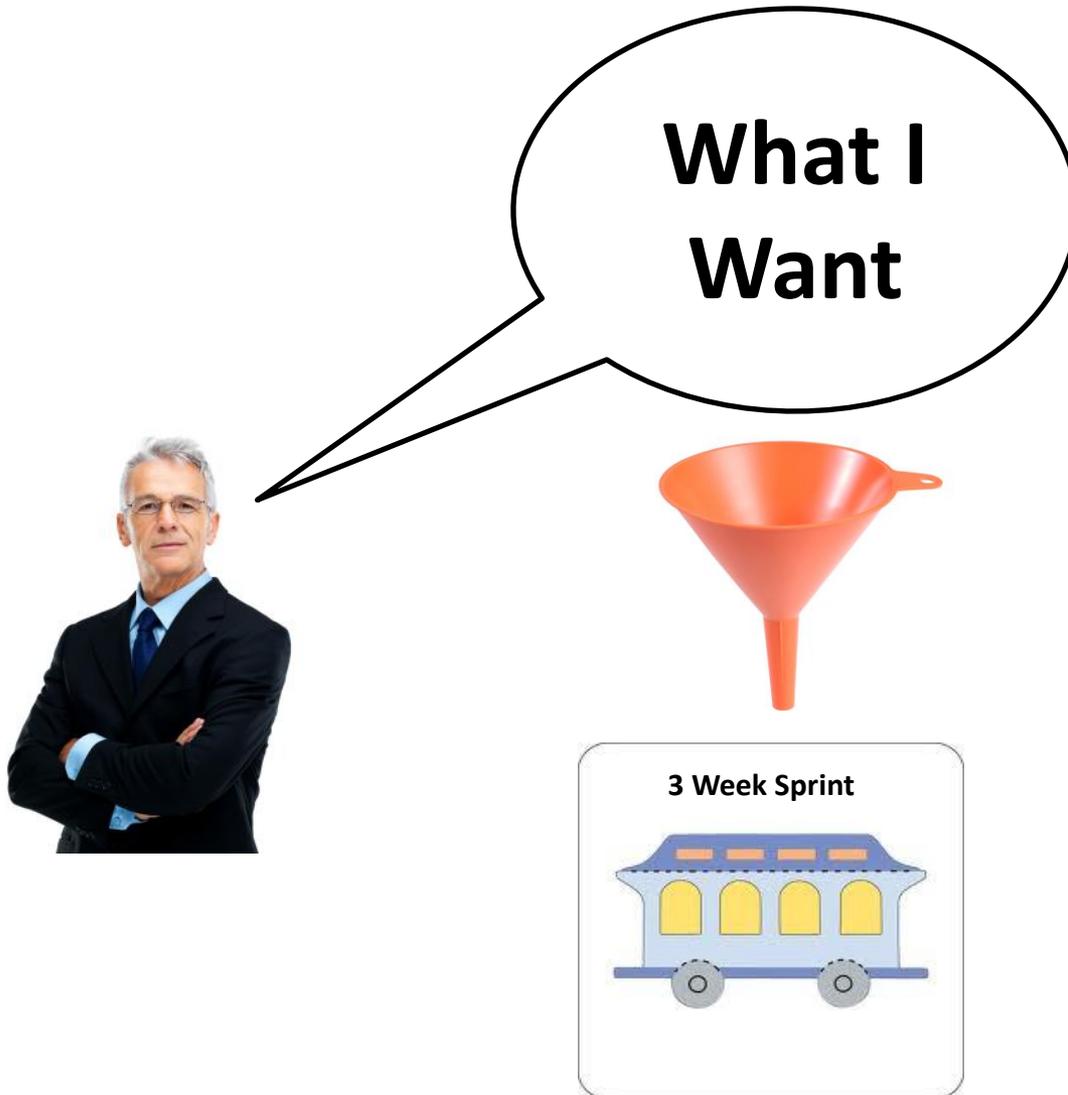
# Complete a Story at a Time – Focus on Customer Value

	Day 1	Day 2	Day 3	Day 4	Day 5
<b>Week 1: Sprint Planning &amp; Start</b>	POs review stories with Dev & QA	<ul style="list-style-type: none"> <li>Devs and QA Estimate.</li> <li>Determine build sequence.</li> <li>Finalize Acceptance Criteria</li> <li>COMMIT!</li> </ul>	Build Write test cases Execute tests Accept Stories Standup	Build Accept Stories Standup	Build Accept Stories Standup
<b>Week 2: BUILD  TEST  ACCEPT</b>	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build Write test cases Accept Stories Standup	Build, test, accept, standup  <b>Meeting for Next Sprint</b>
<b>Week 3 Finalize  Demo  Learn</b>	Build Execute tests Accept Stories Standup	Harden, no new stories to QA	Harden, no new stories to QA	Review with Stakeholders Finalize Stories for Next Sprint Review velocity	Retrospective Update regression test list Update release plan*



# Issue 3 – WYWIWWCT

## What You Want Is What We Commit To



# Issue 3 – WYWIWWCT

## Estimate to make Sure It Fits

Person	Role	Total Time Needed	Total Time Available	Difference	Utilization
Paul	.Net Developer	34 hours	25 hours	9 hours	136%
Diane	User Experience	32 hours	35 hours	+ 3 hours	91%
Keith	Tester	31 hours	31 hours	0 hours	100%
Sanjeev	Ajax Developer	34 hours	37 hours	+ 3 hours	92%

# Issue 4 – Lost Resources



# Issue 4 – Lost Resources

- You can track the losses to explain at the sprint review
  
- Better
  - Dedicate folks to support
  - Establish support teams
  - Back out capacity for support
  
- When crazy, abort and go to Kanban



# Issue 5 – Allowing Disturbances

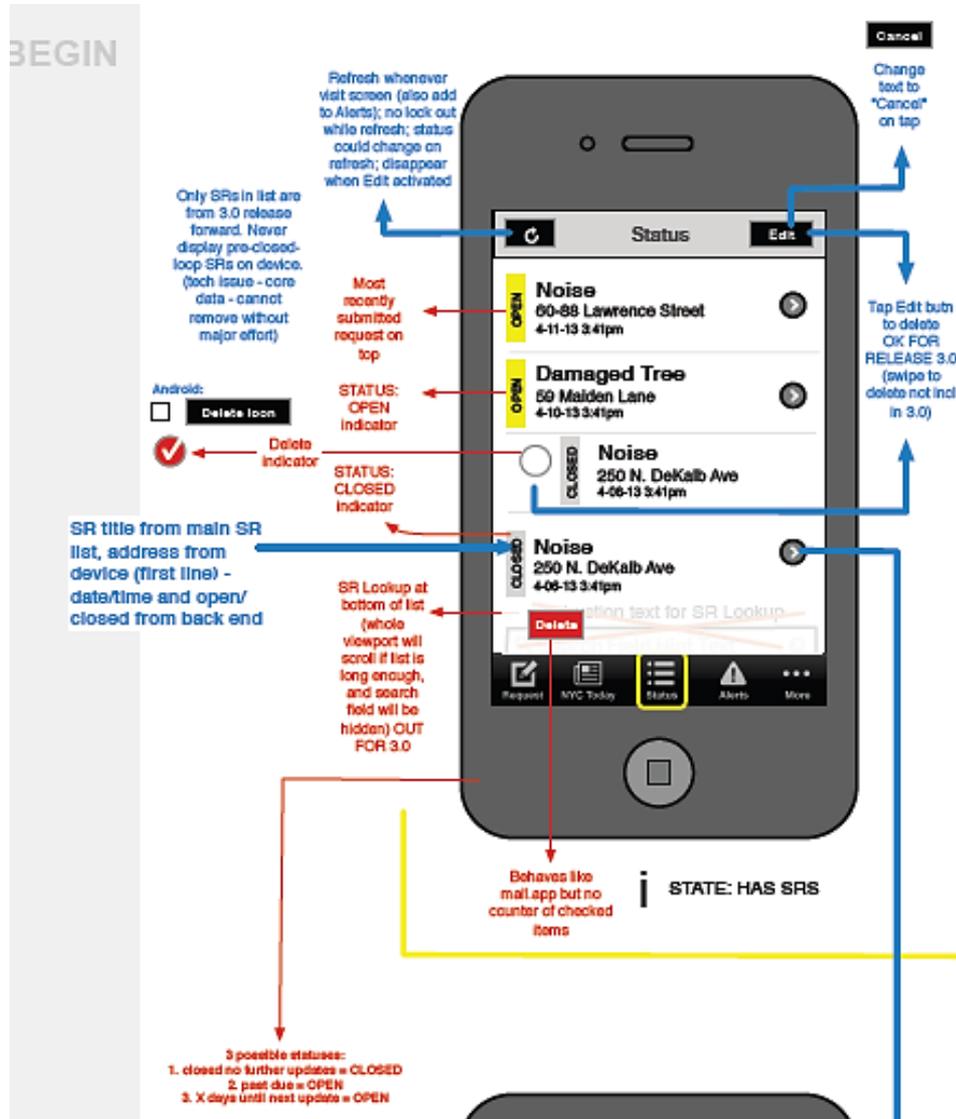


# Issue 5 – Allowing Disturbances

- Agile is all about collaboration and interaction
- However, code does not get written when you are interacting
- Developers need moments of protection and isolation during a sprint



# Issue 6 – No Pre-Work



# Issue 6 – No Pre-Work

- Pre-Work before a sprint begins is not “waterfall”
- It helps jumpstart a sprint when developers are available
- Great for environments with Business Analysts

	SPRINT 1			SPRINT 2			SPRINT 3		
Preweek	Week 1	Week 2	Week 3	Week 1	Week 2	Week 3	Week 1	Week 2	Week 3
prework	plan								
	build								
			complete						
			prework	plan					
				build					
						complete			
						prework	plan		
							build		
									complete

# Issue 7 – Stories too Ambiguous For Coding

## **Story: Account Holder withdraws cash**

As an Account Holder

I want to withdraw cash from an ATM

So that I can get money when the bank is closed

# Issue 7 – Stories too Ambiguous For Coding

- We have to get into the details to code and test
- For those of you with *use case* experience, very similar
- Happy path, exceptions, alternatives

## Story: Account Holder withdraws cash

As an Account Holder  
 I want to withdraw cash from an ATM  
 So that I can get money when the bank is closed

### Scenario 1: Account has sufficient funds

Given the account balance is \ \$100  
 And the card is valid  
 And the machine contains enough money  
 When the Account Holder requests \ \$20  
 Then the ATM should dispense \ \$20  
 And the account balance should be \ \$80  
 And the card should be returned

### Scenario 2: Account has insufficient funds

Given the account balance is \ \$10  
 And the card is valid  
 And the machine contains enough money  
 When the Account Holder requests \ \$20  
 Then the ATM should not dispense any money  
 And the ATM should say there are insufficient funds  
 And the account balance should be \ \$20  
 And the card should be returned

### Scenario 3: Card has been disabled

Given the card is disabled  
 When the Account Holder requests \ \$20  
 Then the ATM should retain the card  
 And the ATM should say the card has been retained

### Scenario 4: The ATM has insufficient funds

# Questions?



[gssolutionsgroup.com](http://gssolutionsgroup.com)

[greg@gssolutionsgroup.com](mailto:greg@gssolutionsgroup.com)

(206) 854-9229