Effective Agile Requirements

Richard Hundhausen
Consultant/Trainer
Accentient, Inc.

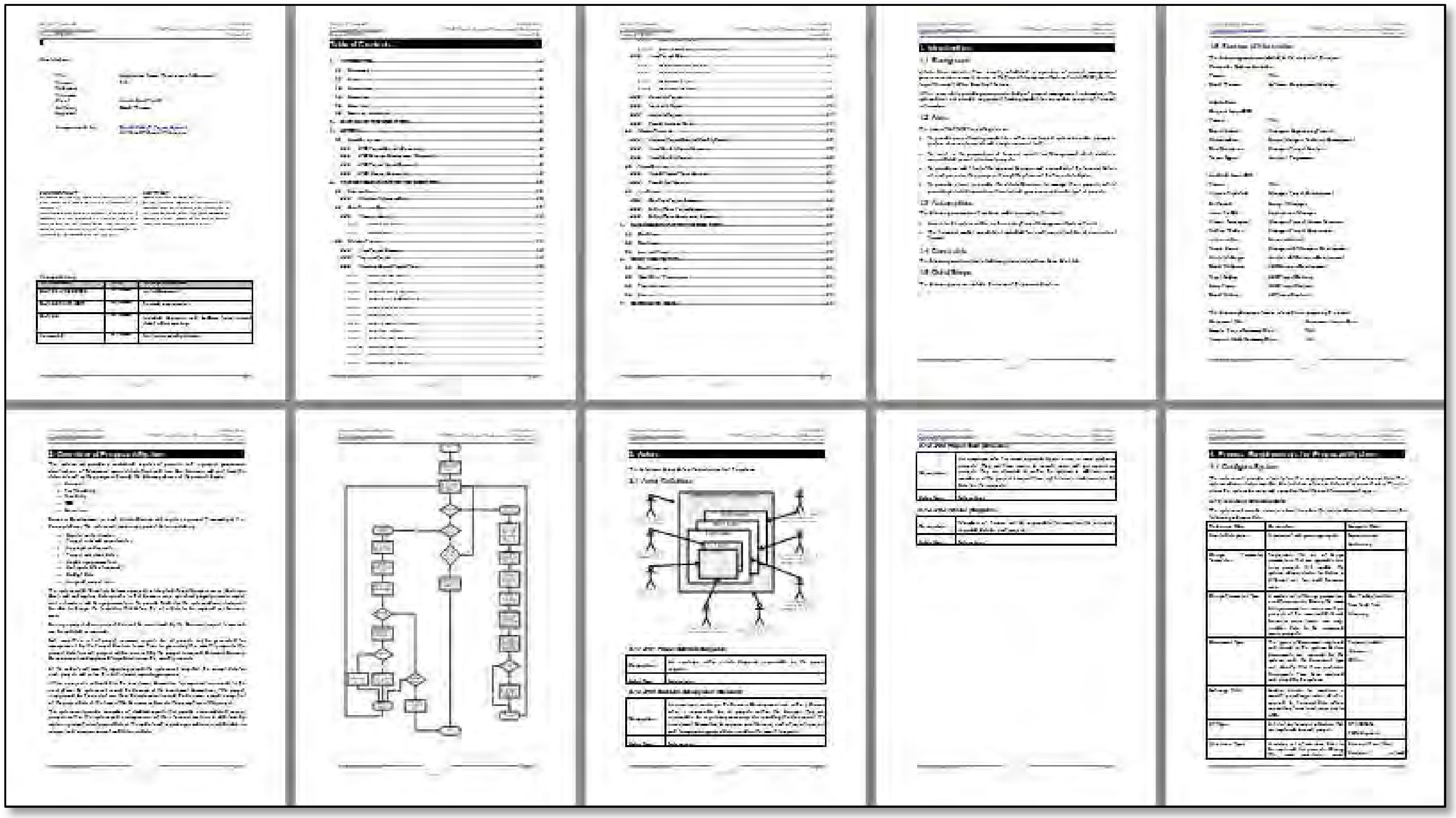
About Me

- From Boise, Idaho, USA
- President of Accentient
- Microsoft Regional Director
- Microsoft MVP (Visual Studio ALM)
- Professional Scrum Developer
- Professional Scrum Trainer
- Author of books and courses
- richard@accentient.com

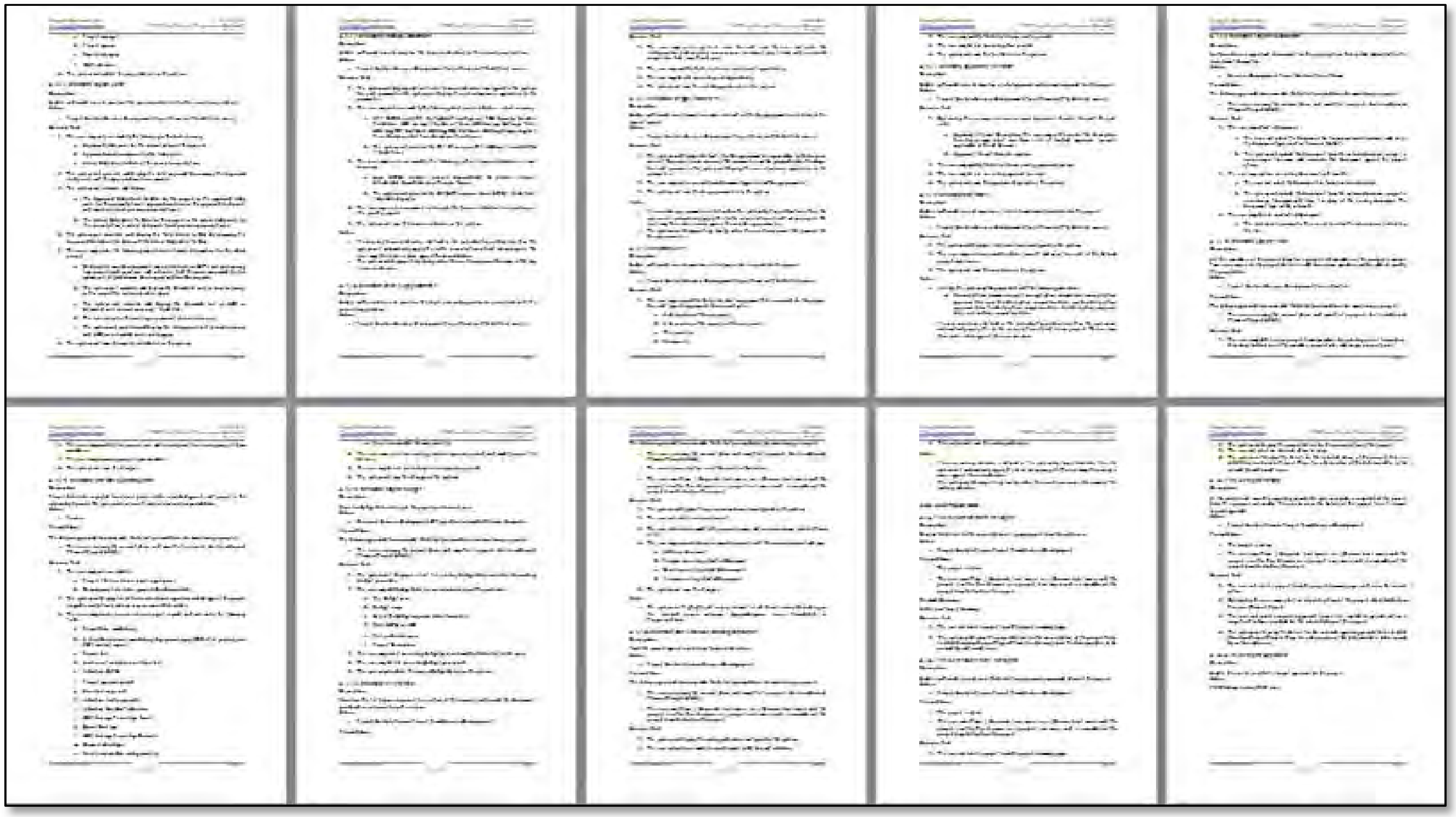
@rhundhausen

Microsoft
Professional Scrum Development with Microsoft Visual Studio 2012
Richard Hundhausen

Here’s a typical requirements document …



Is anyone going to actually read this?



Requirements or *Specifications*?

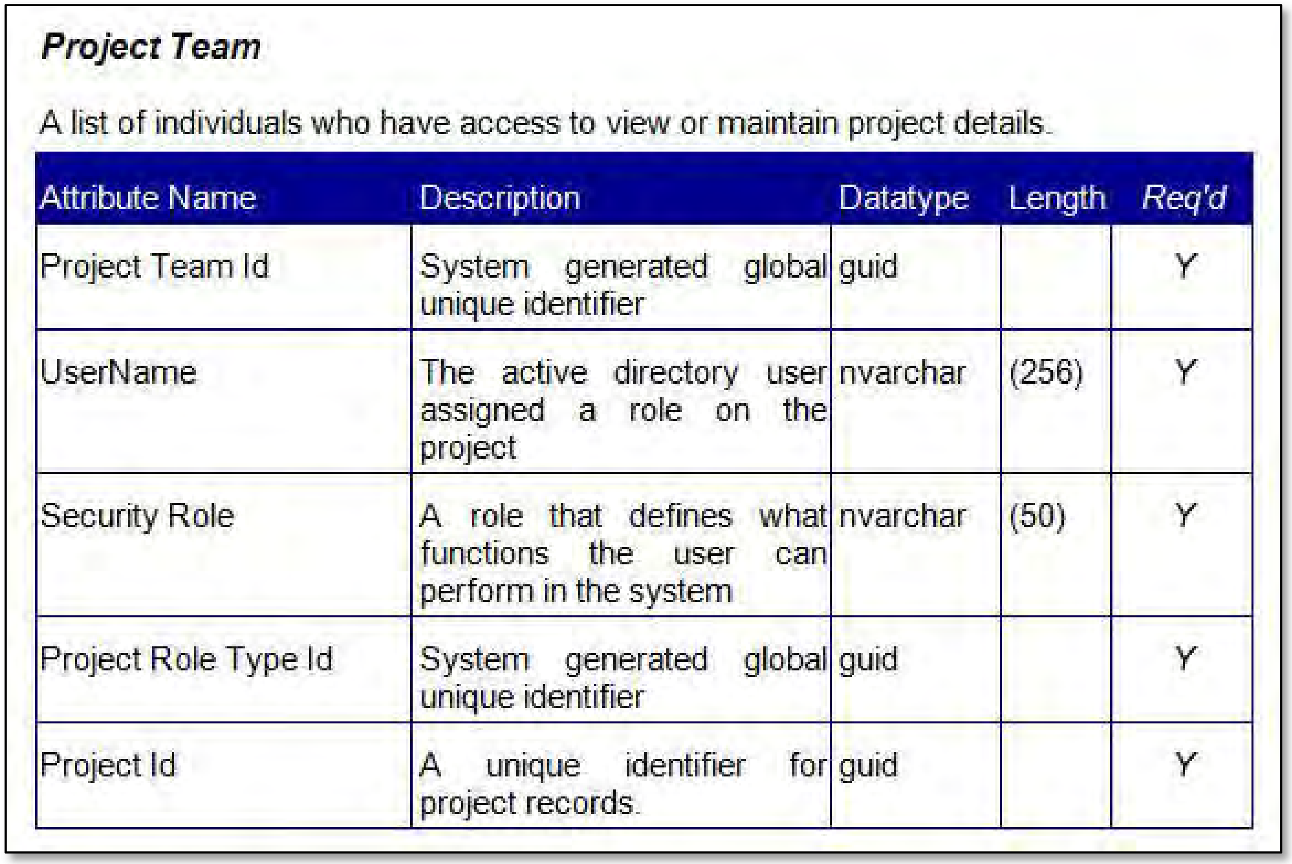
What’s the difference?

- Requirement => Needs & Wants “what”
- Specification => Implementation “how”

  -> Often detached from need

- Users want us to deliver on Needs & Wants
- Tip: Think of Requirements as ‘Desirements’

Usually you get a bit of both …



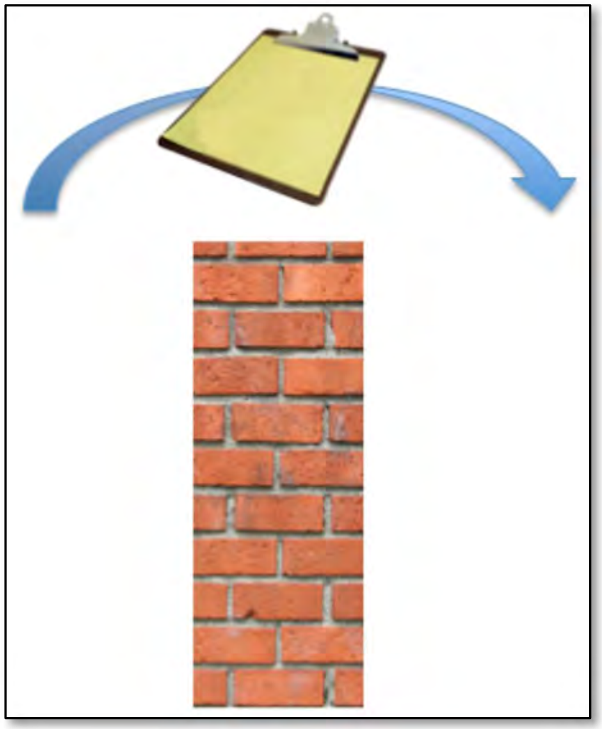
Project Team

A list of individuals who have access to view or maintain project details.

Attribute Name	Description	Datatype	Length	*Req'd*
Project Team Id	System generated global unique identifier	guid		*Y*
UserName	The active directory user assigned a role on the project	nvarchar	(256)	*Y*
Security Role	A role that defines what functions the user can perform in the system	nvarchar	(50)	*Y*
Project Role Type Id	System generated global unique identifier	guid		*Y*
Project Id	A unique identifier for project records.	guid		*Y*

Specifications …

- Are ambiguous
- Are authoritative
- Suppress conversation
- Suppress real discovery



Requirements come in many flavors …

Action
Workstream
story
Theme
Defect
Test
Objective
Scenario
Bug
Use
case
Journey
Feature
Epic
Initiative
User
Behavior
Workflow
Funtionality

Enter the User Story

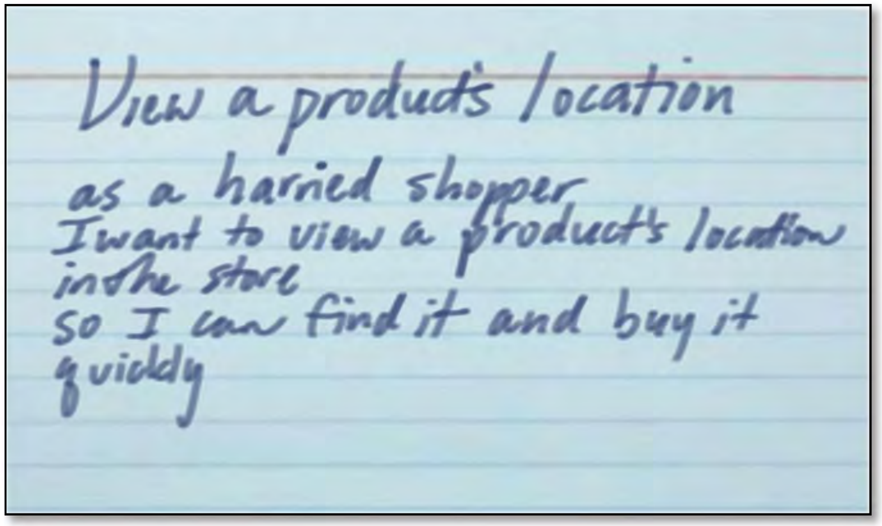
A User Story is …

- A need from the user’s perspective
- Barely sufficient to identify the requirement
- A planning item – a token for a conversation
- Incomplete – a deferred conversation
- Ideally written by anyone

User Story format != a User Story

<TITLE>

- As a <user>
- I want <something>
- So I can <need/reason>



View a product's location

as a harried shopper
I want to view a product's location
in the store
so I can find it and buy it
quickly

Three C’s

- Card
  - The user story (historically written on a card or sticky)
- Conversation
  - Talking with users and customers
  - An *exchange* of thoughts, opinions, and feelings
- Confirmation
  - Acceptance criteria or tests

http://bit.ly/1PqFBuS

User Stories Gain Detail Over Time

- Start with a title
- Add a concise description

```
    As a [type of user],
    I want [some goal],
    so that [some reason]
```

Remember — that's just a thinking template. No need to write all your stories this way.

- Add other relevant notes or sketches
- Add acceptance criteria

Stories need Acceptance Criteria

- Must be objectively verifiable statements
  - Describes a user story’s “behavior”
  - Functional, non-functional, performance, etc.

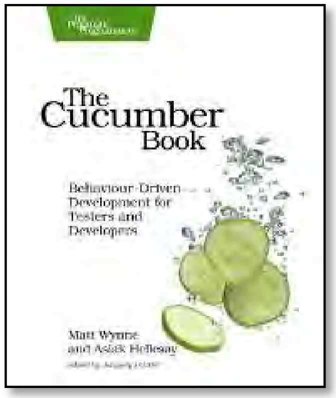
- Examples:
  - Editor tools comply with site design
  - Two second or less for query and page response

Write Acceptance Criteria as Tests

```
Given I have a valid account
 and my balance is $100
 and the ATM has enough cash
When I withdraw $80
Then I should receive $80 in cash
 and my account balance should be $20
 and my card should be returned
```

BDD? Gherkin? What the Cuke?!

- Behavior Driven Development
  - The behavior you want drives what gets developed, not a specification document
- Given/When/Then is "Gherkin" syntax
  - JBehave, Rspec, Cucumber
  - SpecFlow, MSpec


The Cucumber Book
Behaviour-Driven Development for Testers and Developers
Matt Wynne and Aslak Hellesøy

specflow
Cucumber for .NET

www.specflow.org

When are we ready to code?

The Product Backlog

- A collection of user stories for a software product is referred to as the *product backlog*
- The product backlog is ordered in a way that the most valuable (and ready) items are at the top

INVEST in Good User Stories

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

http://bit.ly/1TxK01e

Definition of Ready

- A team’s explicit and visible criteria that a user story must meet prior to being accepted into the next sprint
  - Typically based on INVEST
  - Analogous to the definition of “done“

http://bit.ly/1ZLPn59

What’s the ROI?

Value = The “R”

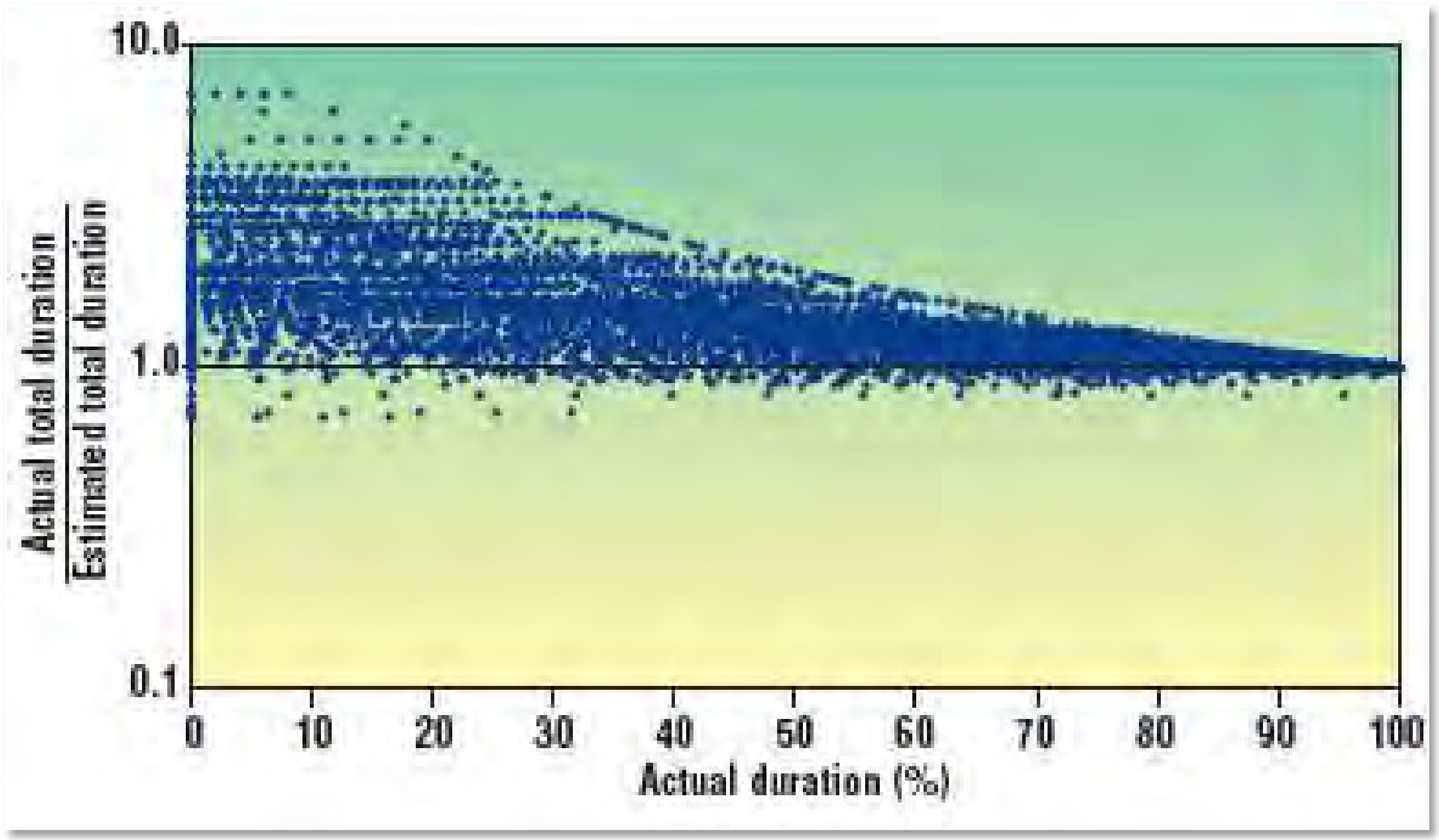
- Unfortunately, “value” is hard (impossible?) to estimate or measure
  - Market value? Risk reduction? Capability building?
  - It’s one of the “unicorn” metrics (along with *productivity* and *technical debt*)
- That said, there are practices for assessing value
  - MoSCoW, Prioritization Poker, $100 method
  - Involve your stakeholders

http://bit.ly/1sIoOIx

Size/Effort = The “I”

- The “cost” of an item is estimated by the development team – those doing the work
- Estimates should be in an abstract unit of measure
  - Hours, days, or $ will imply a commitment, plan, or budget
- Planning Poker® and affinity estimation practices

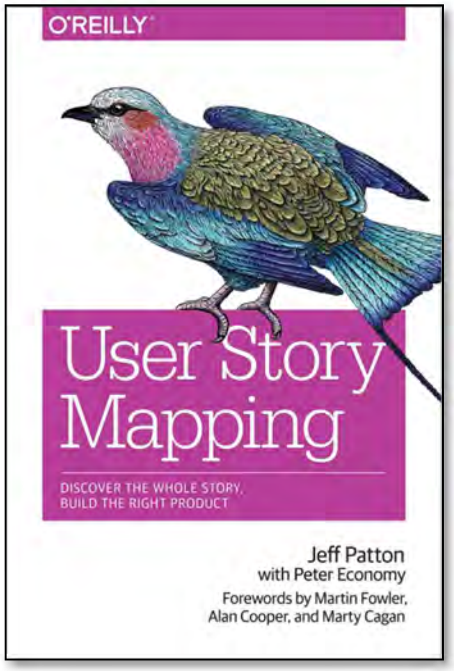
Estimation becomes more accurate over time



Source: IEEE Software May/June ‘06, Todd Little

A one-dimensional product backlog can’t really tell us the whole story

Enter the Story Map


O'REILLY

User Story Mapping

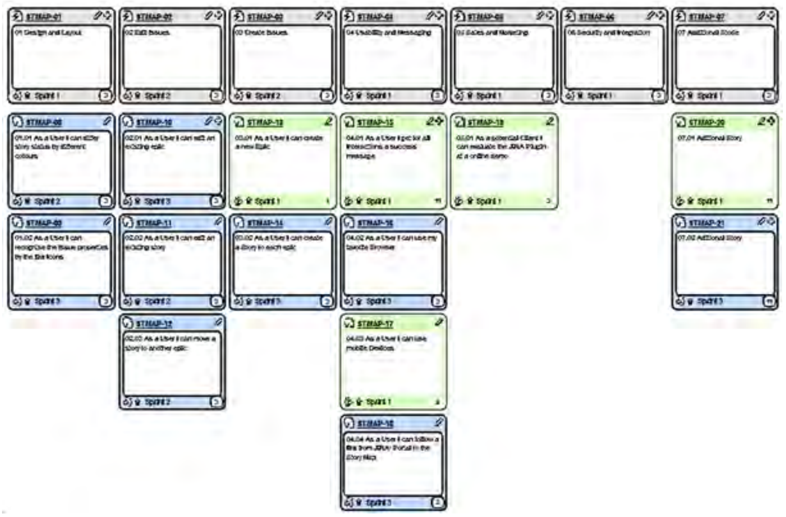
DISCOVER THE WHOLE STORY, BUILD THE RIGHT PRODUCT

Jeff Patton
with Peter Economy

Forewords by Martin Fowler, Alan Cooper, and Marty Cagan

The Story Map

- A two-dimensional backlog
  - At the top are large user stories sometimes called “epics”
  - Below the epics are the actual story cards ordered by the product owner
  - The first horizontal row typically contains the essential capabilities



http://bit.ly/1miM3Y2

Story Maps …

- Help map goals to activities to user stories
- Show the relationships of larger stories to their child stories
- Help confirm the completeness of your backlog
- Provide a useful context for ordering the backlog
- Help plan releases in complete and valuable slices of functionality

Retrospective …

- Strive for agile software requirements
  - Ensure your backlog contains “what’s” and not “how’s”
  - User stories make for lightweight requirements
  - User stories evolve over time
  - Ensure the product backlog is regularly refined
  - Development team estimates and even re-estimates
  - Do all of the above at the last responsible moment

Remember …

richard@accentient.com | @rhundhausen