

# Better User Stories: Have Your Cake and Eat it Too

As who, I want  
what so that why.

Hosts: *Jeff Sutherland*  
*Joel Riddle*



# Who We Are

Scrum Inc. is the Agile leadership company of Dr. Jeff Sutherland, co-creator of Scrum. We are based at the MIT Cambridge Innovation Center, MA.

**CEO Jeff Sutherland** helps companies achieve the full benefits of Scrum leading our comprehensive suite of support services and leadership training:

- Adapting the methodology to an ever-expanding set of industries, processes and business challenges Training (Scrum Master, Product Owner, Agile Leadership, online courses, etc.)
- Consulting (linking Scrum and business strategy, scaling Scrum)
- Coaching (hands-on support to Scrum teams)



**Chief Content Officer JJ Sutherland and Scrum Master Joel**

**Riddle** maintain the Scrum framework by:

- Capturing and codifying evolving best practices (Scrum Guide)
- Conducting original research on organizational behavior
- Publishing (3 books) and productizing ScrumLab



**President Scrum@Hardware Joe Justice** leads our hardware consulting practice:

- Worldwide consulting at leading hardware companies
- 700-800% performance improvement in hardware development
- Builds 100 mpg cars in his garage with help from 500 people in 32 countries



We run our company using Scrum as the primary management framework, making us a living laboratory on the cutting edge of “Enterprise Scrum”

Find out more at [www.scruminc.com](http://www.scruminc.com).

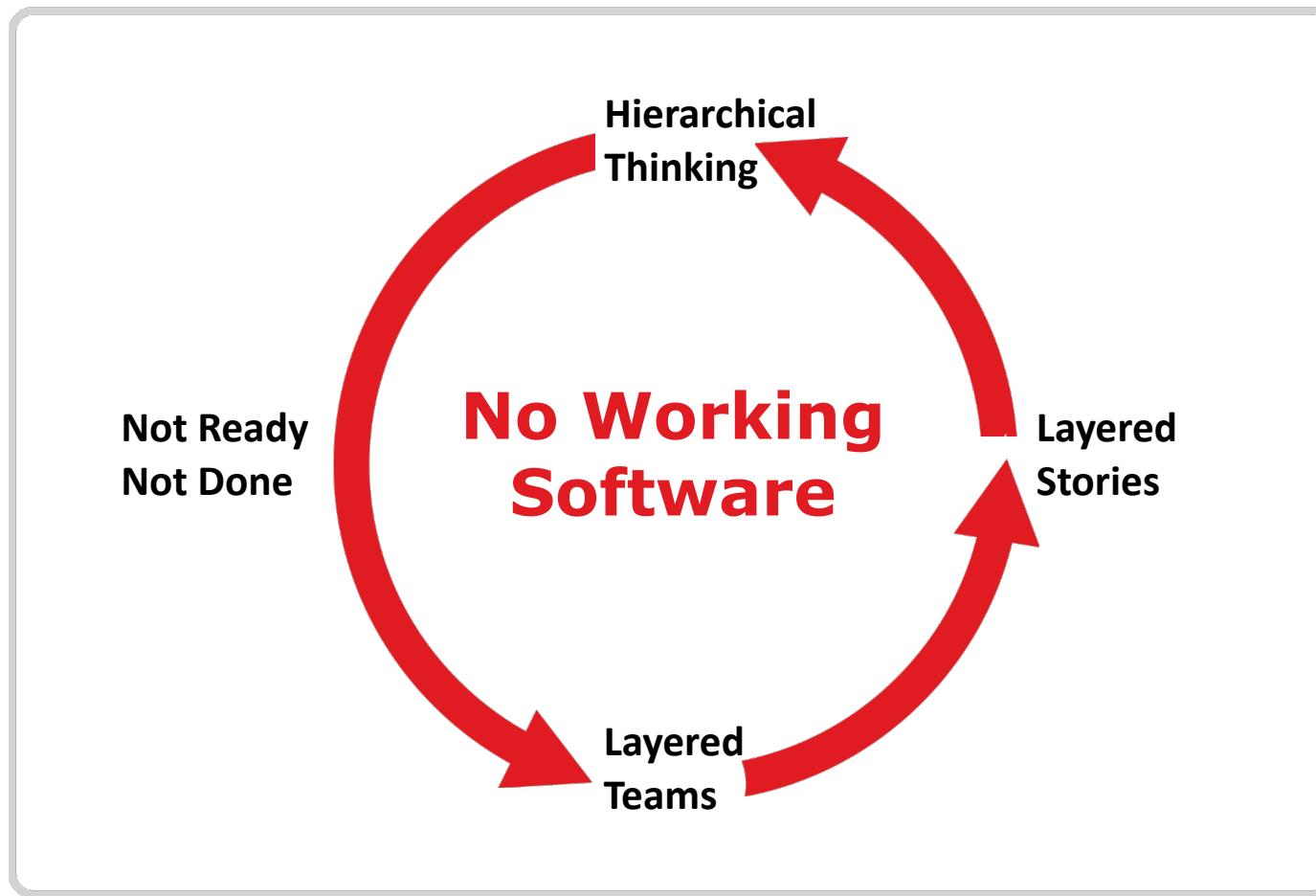
# Agenda

- Discuss what makes a “good” user story, and the importance of good user stories to Scrum
- Introduce “Stacks” and vertical slicing of functionality
- Share four tips for writing good independent user stories
  - Product decomposition
  - Modular architecture
  - User stories, not tasks
  - Sanity check often
- Share several different examples of good user stories in different contexts

# Agile User Stories



# The Four Horseman of the Apocalypse



*Obamacare had no working stories at the beginning, in the middle, or at the end!*

# Symptoms of Bad User Stories

## Wasted Time

- Excessive effort to figure out what is really meant by the story
- Additional research needed before work can start/end
- Time spent waiting for external dependencies to be cleared

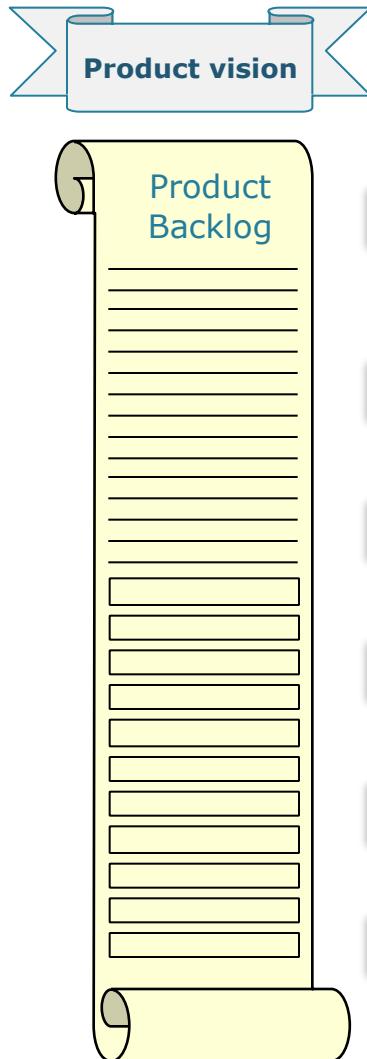
## Product Issues

- Back-end infrastructure built with nothing to show customer
- Building something only to discover it is not what the customer really wanted
- Overly prescriptive stories don't leave room for innovation by the team

## Quality Problems

- Insufficient Definition of Done results in poor product quality
- Sub-components of product developed by different teams don't integrate well
- Over-built features due to lack of clear acceptance criteria cause code bloat and product liability

# User Story Readiness Guidelines



Modified from Bill Wake – [www.xp123.com](http://www.xp123.com)

# User Story Readiness Progression

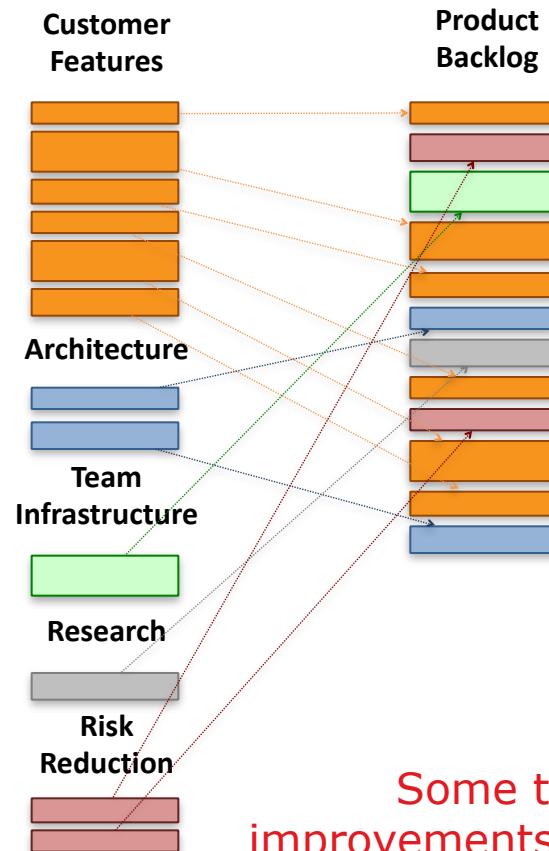


New Card Nursery	<ul style="list-style-type: none"><li>• All inputs accepted</li><li>• <b>Promotion:</b> Product Owner determines this story matches product goals</li></ul>
Elementary School	<ul style="list-style-type: none"><li>• Analysts decompose</li><li>• User experience experts research context</li><li>• Business alignment needs identified</li><li>• <b>Promotion:</b> Matches release goals</li></ul>
Junior High	<ul style="list-style-type: none"><li>• Card details, acceptance criteria, UI pre-work (wireframes, visual and content prototypes)</li><li>• Legal &amp; compliance issues reviewed</li><li>• <b>Promotion:</b> Alignment with key stakeholders on features, functions, and visuals</li></ul>
High School	<ul style="list-style-type: none"><li>• Ready for sprint</li><li>• Candidates for Release Planning/Sprint Planning</li><li>• Minimal refinement expected on core User Experience</li></ul>

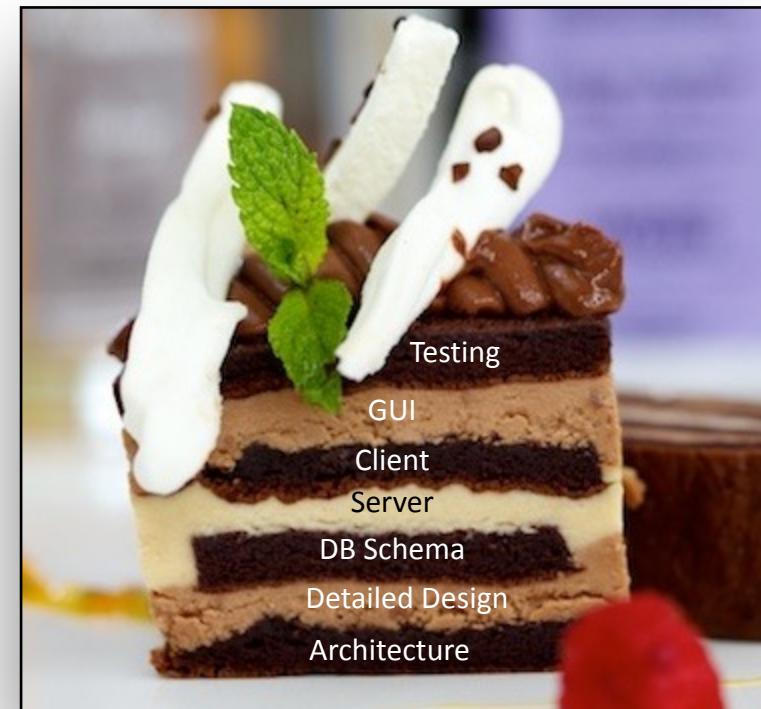


# Not All Backlog Items are User Stories, But All User Stories Should be “Vertical Slices”

Backlog items include everything the team needs to do in one ordered set of activities



Wherever possible, backlog items should deliver complete vertical slices of functionality across work layers



Some teams also choose to include process improvements, bugs and technical debt fixes explicitly as backlog items

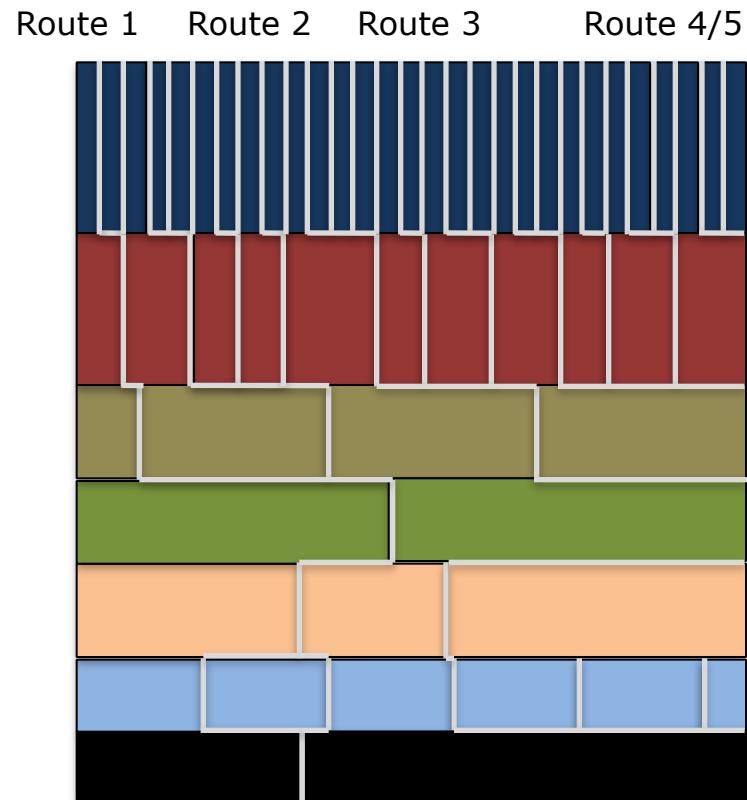
# Breaking the Stack into Independent Stories

**All industries have  
“stacks”**

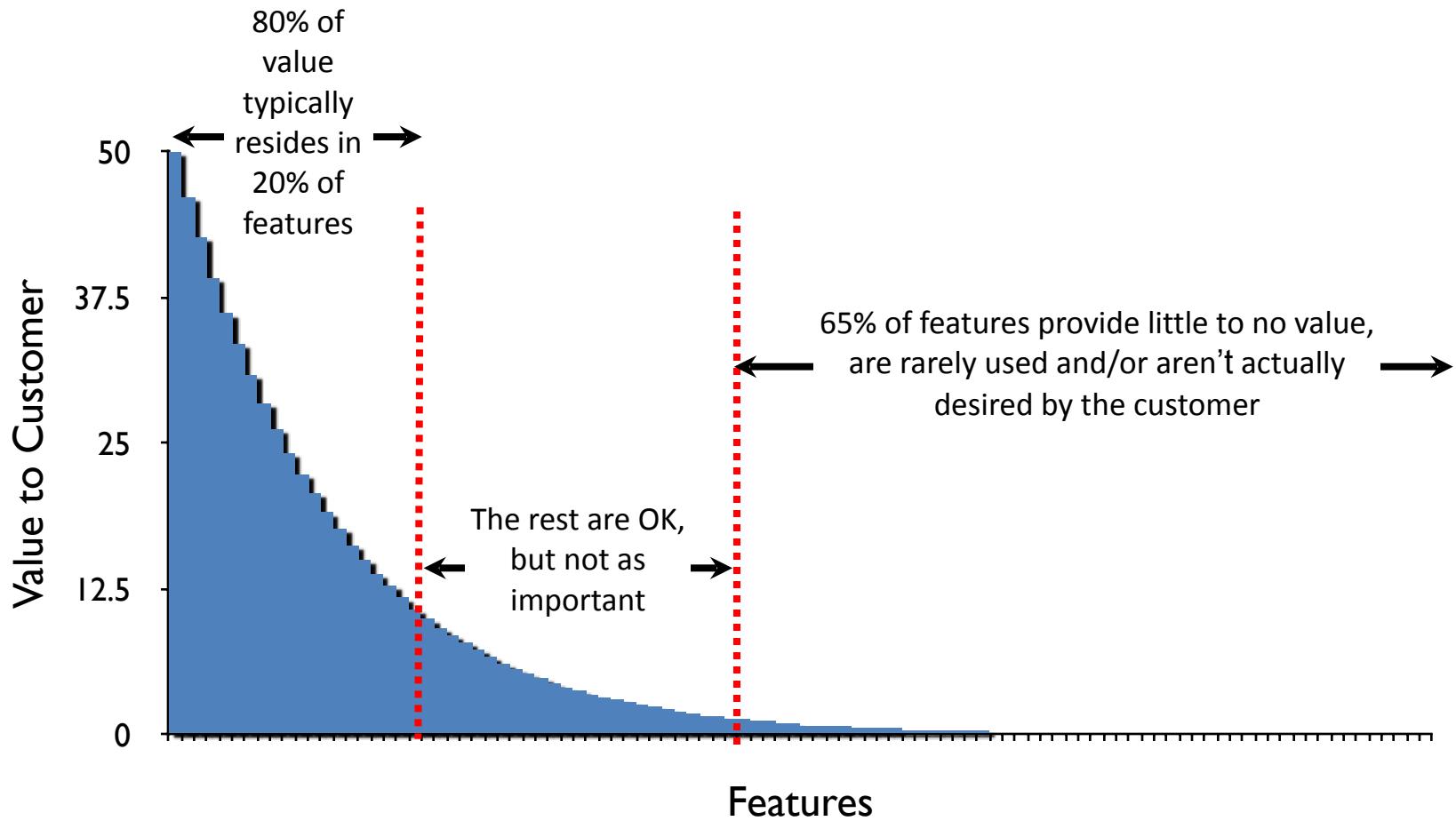
E.g. Air travel industry stack



**A compelling user story  
delivers incremental value  
across stack layers**



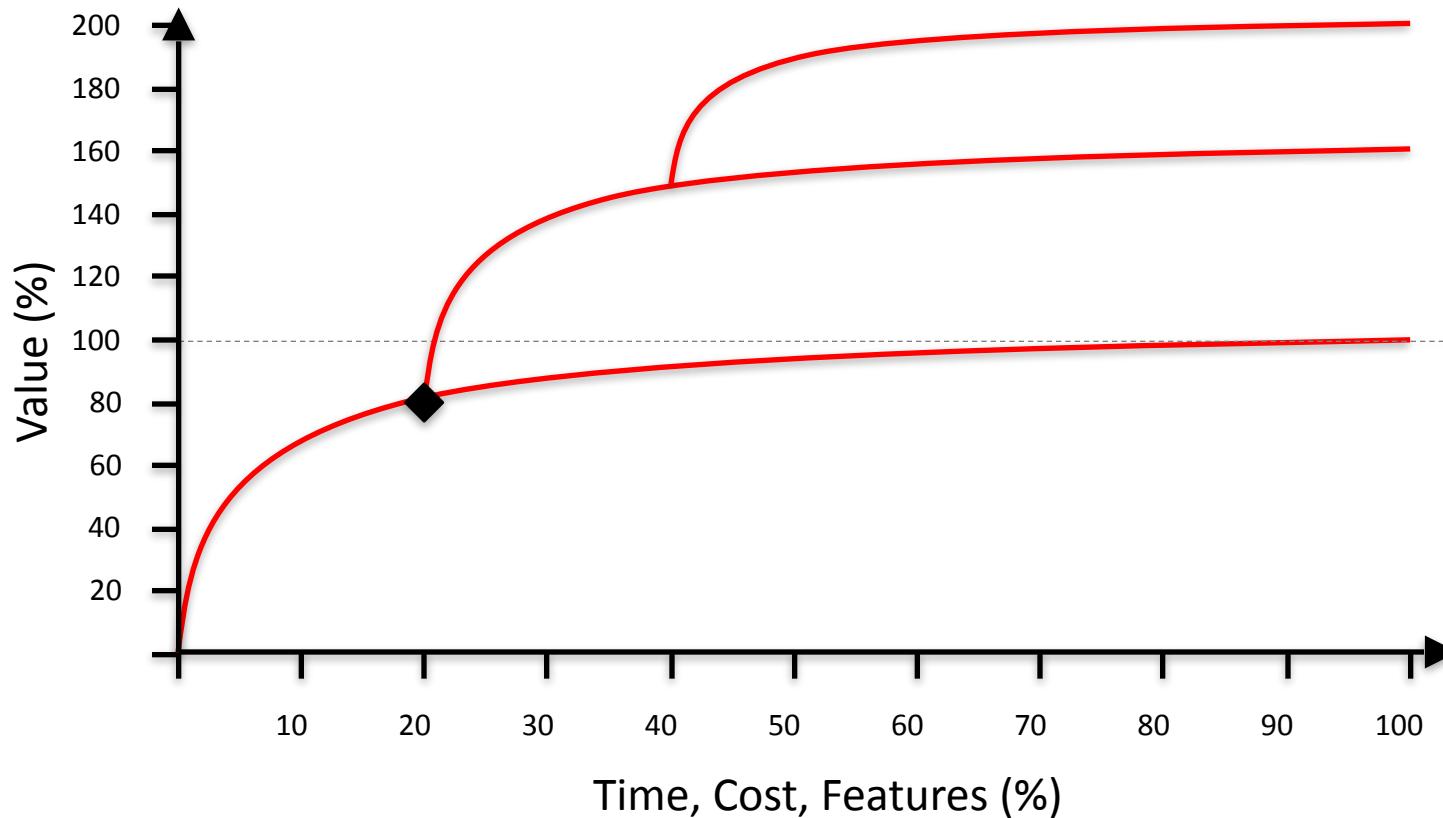
# Not All Features Are Created Equal!



How can you tell ahead of time which features add value and which don't?

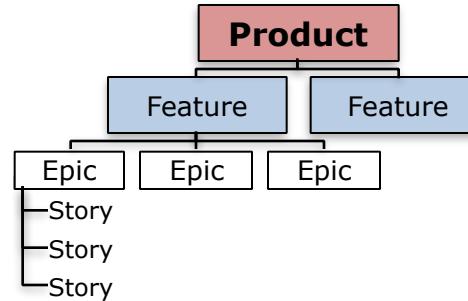
Better

# Delivering Customer Features Incrementally Can Drive Radically Better Value Delivery

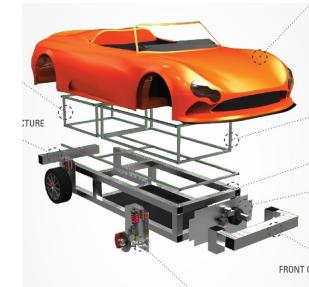


# Four Tips for Writing User Stories as Independent Vertical Slices

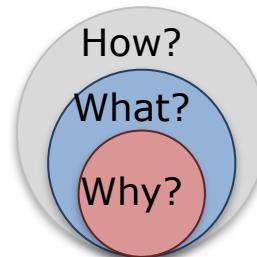
1 Maintain and use clear product decomposition



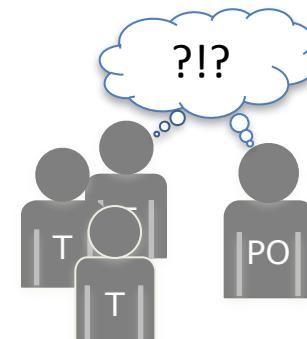
2 Leverage modular/agile architecture as a foundation



3 Write User Stories, not Tasks

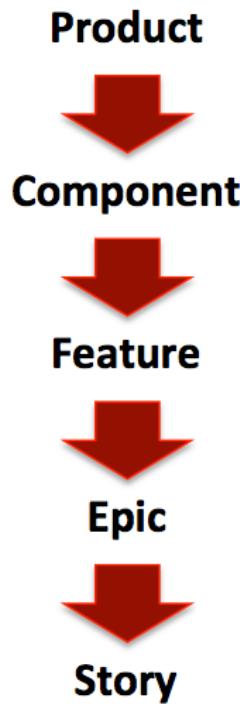


4 Conduct regular vertical slice “sanity checks” on all stories

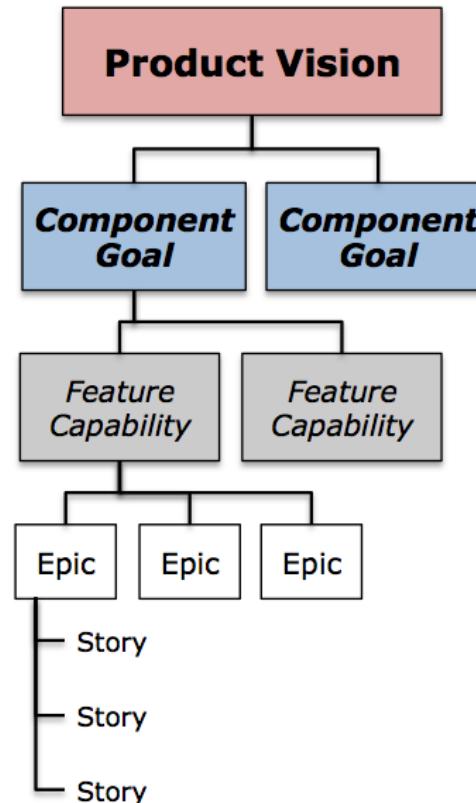


# Maintain a Clear Product Decomposition Hierarchy

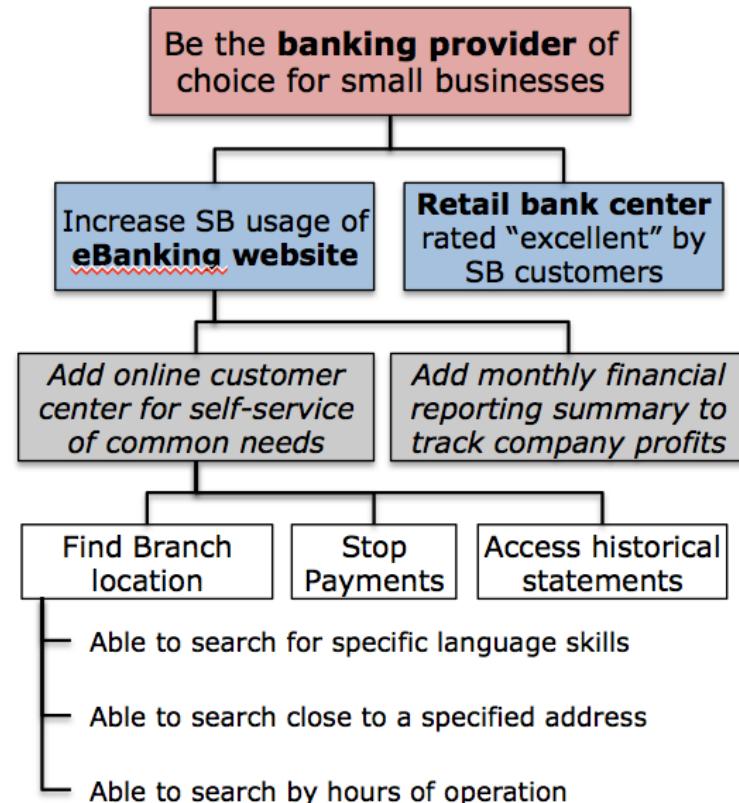
## Decomposition Level



## Conceptual Hierarchy



## Example



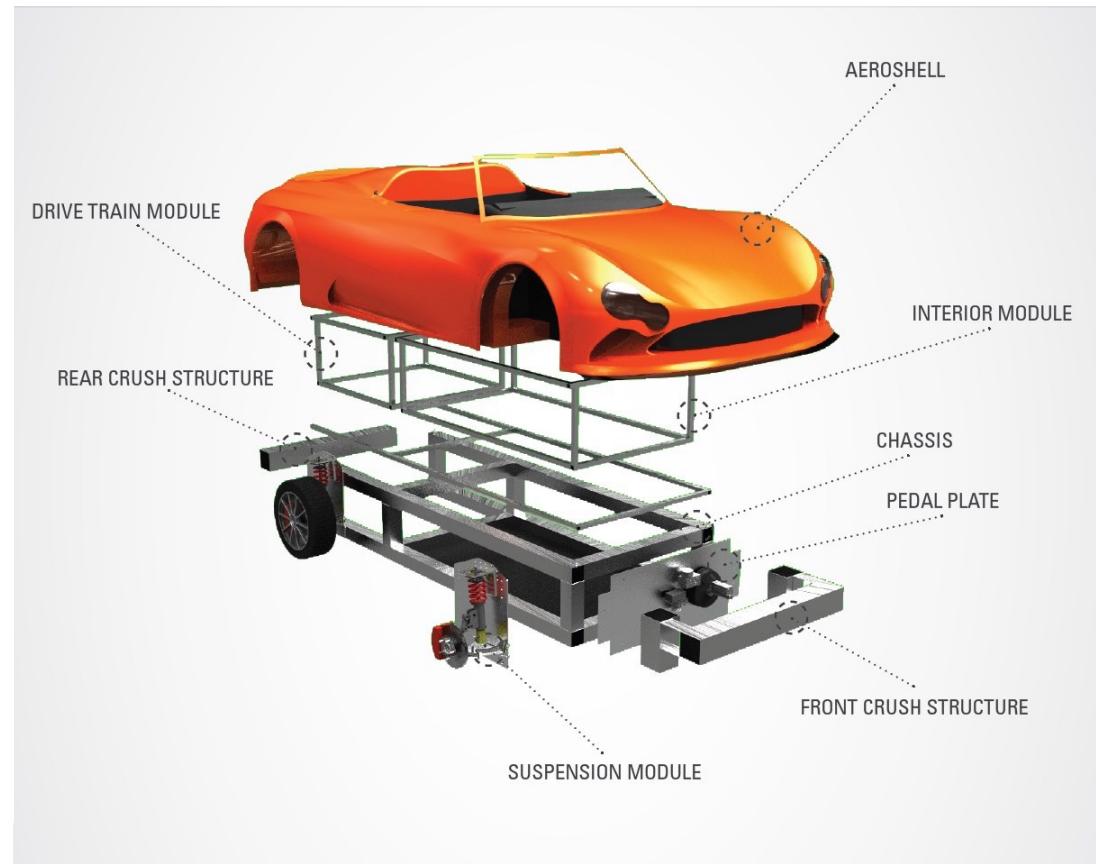
# Modular/Agile Architecture Needs to Support Product Hierarchy!

- Underlying structure is a set of largely independent modules with pre-defined interfaces
- Interfaces remain stable, allowing everything within the module to change without impacting other modules



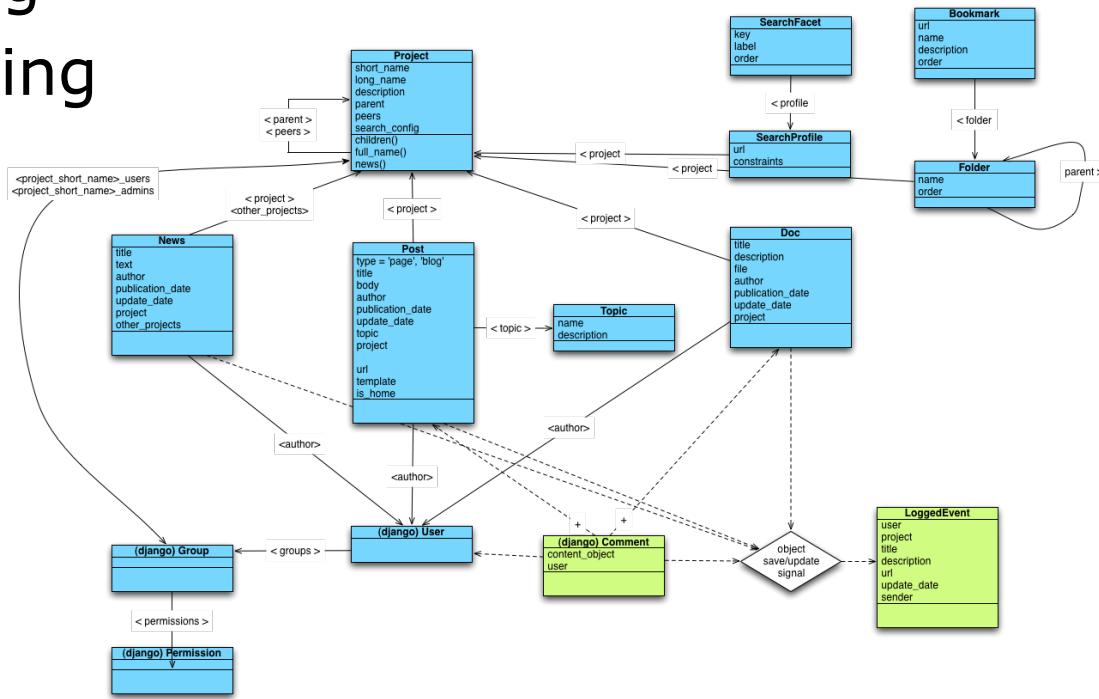
- Enables product design to “emerge” rapidly in response to inspect and adapt cycles
- Also supports re-use of the same module for different contexts

The 8 modules of the Wikispeed Car



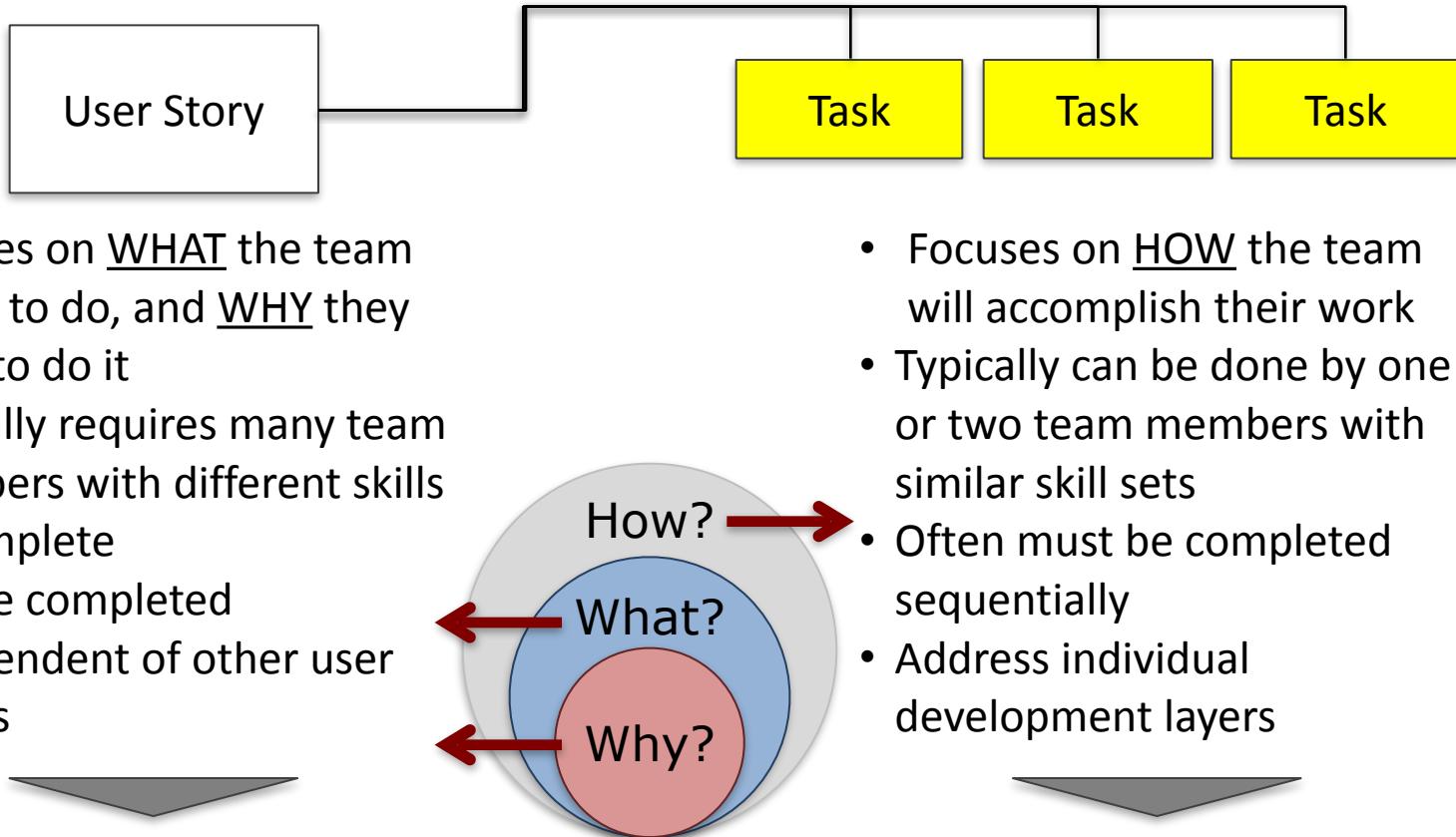
## 2 In Software, “Object Oriented” Modularity Has Been the Norm for a Long Time

- Business components
- Message passing
- Information hiding
- Inheritance
- Polymorphism
- Refactoring



*A type of user needs an object to do something to generate value!*

# Write User Stories, Not Tasks



**Deliver independent customer visible value!**

**Do not deliver independent customer visible value!!**

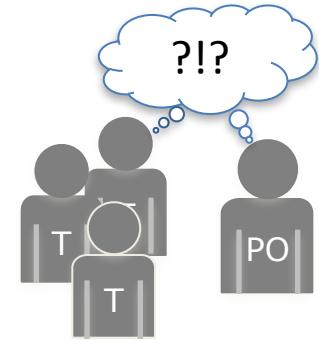
Confusing user stories with tasks unnecessarily limits the team's ability to innovate, accelerate and try new approaches

# Conduct Regular “Sanity Checks”

Despite the best intentions, dependent or task-level stories invariably slip through...

Make time as a team to check stories in the backlog regularly

- E.g. at Sprint Planning or Backlog Refinement



- Customer visible value** – does every story result in customer visible value? (customer not necessarily just an external user)
  - If no then story probably isn't independent
- Swarming** – does this story require multiple people to complete?
  - If no then it probably isn't a complete vertical slice
- External Dependency** – Is this story free from dependence on other stories or groups outside the team?
  - If no then story probably isn't independent
- Test Driven Development** – Does this story have clear and testable acceptance criteria?
  - If no then it probably isn't a complete vertical slice

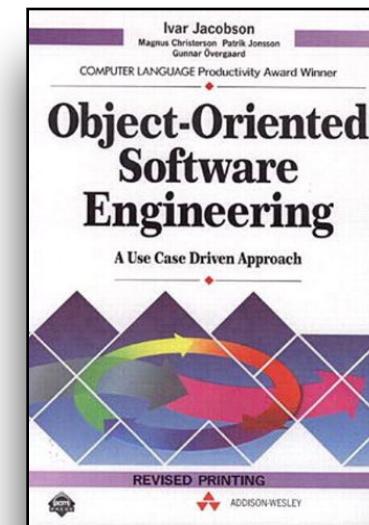
# Example 1: Books and Beyond

- We are building an application for a business that sells products such as books, movies, music, and greeting cards. Assume a physical store.
- Your Product Owner has a story: **As a customer, I want to buy a product so that I can enjoy using it!**
- This story is a huge epic. The team needs to work with the product owner to split it.



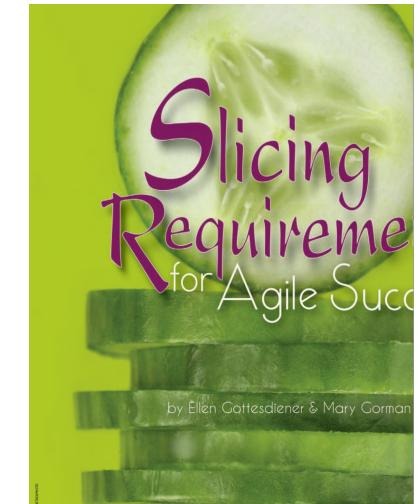
# Where Do We Start?

- What is the first story you would implement?
- Get it ready:
  - Immediately actionable
  - Negotiable
  - Valuable
  - Estimable
  - Testable
- Any non-functional requirements?



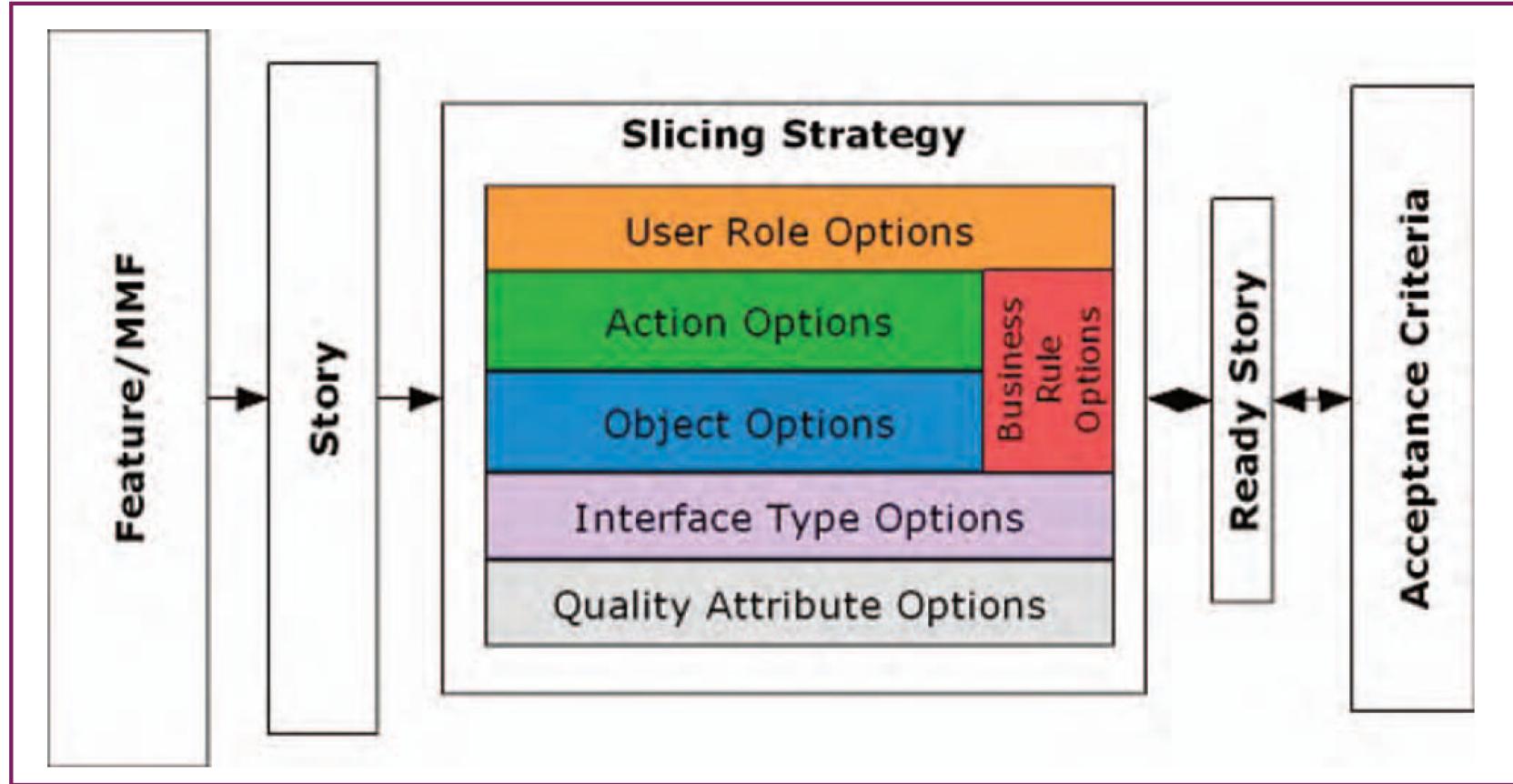
# Slicing User Story Options Based on Value

- Slicing Requirements for Agile Success
  - Ellen Gottesdiener and Mary Gorman. Better Software Jul/Aug 2010
- Inspired by:
  - Chris Matts and Olav Masson on real options and feature injection
  - Bill Wake and others on story splitting
  - Jeff Sutherland and others on ready requirements
  - Dean Leffingwell on lean backlog
  - Mike Cohn on minimizing team handoffs



[http://www.ebgconsulting.com/Pubs/Articles/SlicingRequirementsForAgileSuccess\\_Gottesdiener-Gorman\\_August2010.pdf](http://www.ebgconsulting.com/Pubs/Articles/SlicingRequirementsForAgileSuccess_Gottesdiener-Gorman_August2010.pdf)

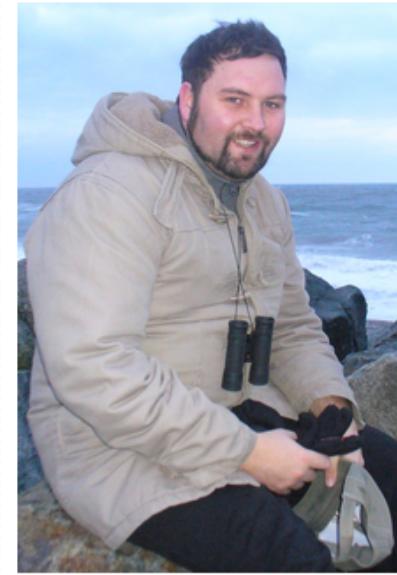
# The Six Slicing Elements of a User Story



# User Role Options: Types and State

- What are possible user types?
  - Individual Buyer
  - Corporate Buyer
  - Club Member Buyer
  - Employee Buyer
- What are possible user roles?
  - New
  - Existing
  - Anonymous
  - Archived
- What combination yields the highest immediate value?
  - **Individual Anonymous Buyer**

Charlie Jones (Online Student - Adult Learner)



"I want technology to stay out of my way."  
[wiki.fluidproject.org](http://wiki.fluidproject.org)

# Buyer Action Items

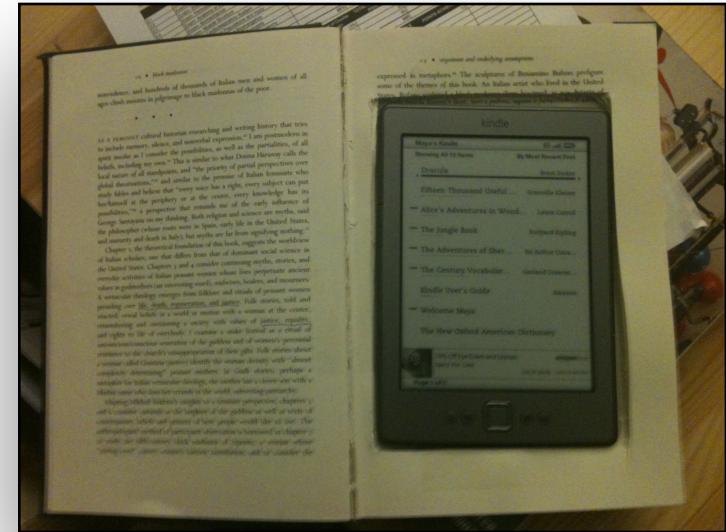
- To identify all possible buyer actions, consider “I want to buy a product.”
- Ask the Product Owner what typically happens for an Individual Anonymous Buyer.



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

# Buyer Action Items

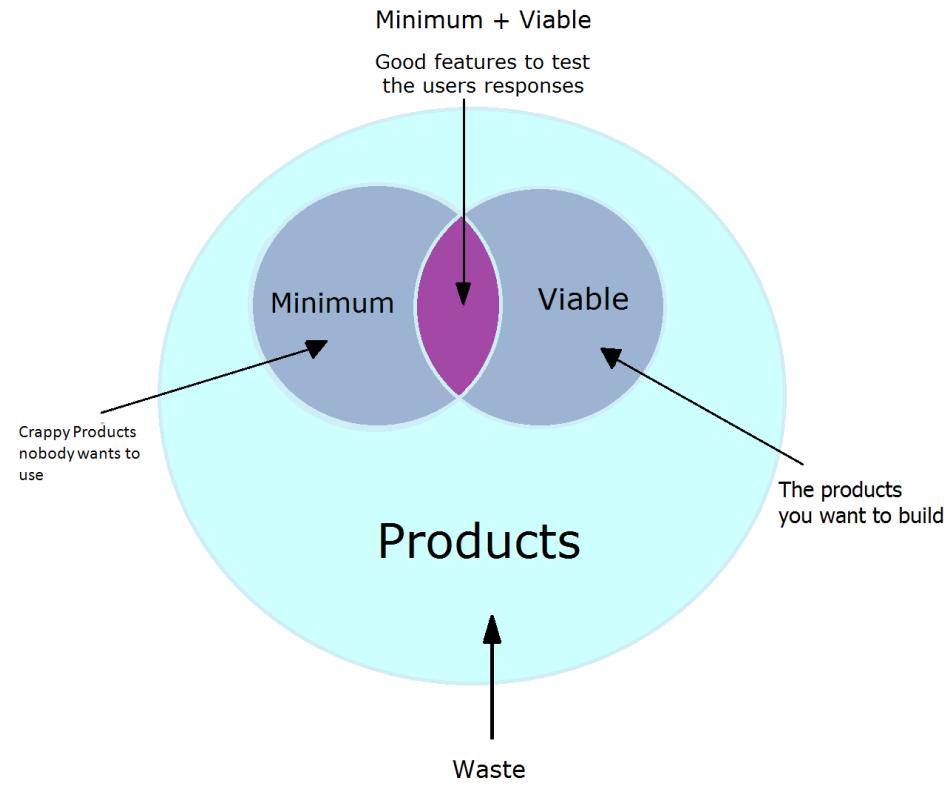
- To identify all possible buyer actions, consider “I want to buy a product.”
- Ask the Product Owner what typically happens for an Individual Anonymous Buyer.
  - **Verify product cost**
  - **Calculate tax amount**
  - **Calculate total purchase amount**
  - **Apply discount**
  - **Apply wrapping fee**
  - **Arrange for shipping**
  - **Secure payment**
  - **Adjust inventory**
  - **Generate receipt**
  - **Post payment to accounts receivable**



[library.barnard.edu](http://library.barnard.edu)

# What are the Minimum Requirements for the Next Delivery Cycle?

- **Verify product cost**
- Calculate tax amount
- **Calculate total purchase amount**
- Apply discount
- Apply wrapping fee
- Arrange for shipping
- **Secure payment**
- Adjust inventory
- **Generate receipt**
- Post payment to accounts receivable



# Data Option Types and States

- What are product types?
- What are payment types?
- What are receipt types?



**PayPal**™

# Data Option Types and States: Select for Value

<i>Product Type Options</i>	<i>Payment Type Options</i>	<i>Receipt Type Options</i>	<i>Book State Options</i>
✓ Book	✓ Cash	✓ Cash receipt	✓ New
CD	Credit card	Credit card receipt	Used
DVD	PayPal		
Gift card	Purchase order		
Greeting card			
Electronic book reader			

# Sliced and Diced Story (so far)

- As an **individual anonymous buyer**, I want to **buy a new book with cash and receive a cash receipt**.



# Step 4: Business Rule Options

## *Business Rule Options*

- ✓ Payment currency must be specific to purchase location  
Cash payment denomination amount must not be greater than ...
- ✓ Payment change amount is calculated as ...  
Receipt bar code is designed using ...



# Exercise Step 5: Interface Type Options

## *Book Interface Type Options*

- Scanner (hardware)
- ✓ Keyed in data (UI)

## *Cash Payment Interface Type Options*

- Cash machine (hardware)
- ✓ Keyed in data (UI)

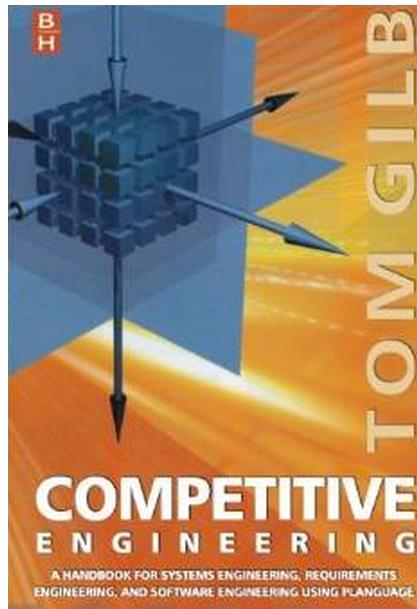
## *Cash Receipt Interface Type Options*

- ✓ Printed in store (report)
- Faxed (system to system)
- Emailed (system to system)



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

# Quality Attribute Options



One quality attribute we need for our sample story is the response time for printing the receipt. Borrowing from Gilb's Planguage tags, you can specify response-time requirements as follows:

Tag: ResponseTime.CashReceiptPrintLaunch

Scale: Seconds

Meter: Elapsed time between pressing "Receipt" to the start of printing

Minimum: No more than 7 seconds

Plan: 4 seconds

Wish: 2 seconds

Alternatively, you can write your story's quality attributes on the back of the user story card (or in your backlog management tool)—for example, “Cash receipt begins printing within four seconds of pressing the Receipt key.”

# Sliced Story

- **Immediately Actionable**
- **Negotiable**
- **Valuable**
- **Estimable**
- **Sized to fit**
- **Testable**



“As a *customer*, I want to *buy a product ...*” Using the story-slicing technique, we successively sliced it into these high-value options:

User role type and state: Individual, anonymous buyer  
Actions: Verify product price, calculate total purchase amount, secure payment, generate receipt

Objects (type and state): New book, cash payment method, cash receipt

Business rules: Payment currency must be specific to purchase location, payment change amount is calculated as ...

Interfaces:

Book interface type: Keyed in data (UI)

Cash payment interface type: Keyed in data (UI)

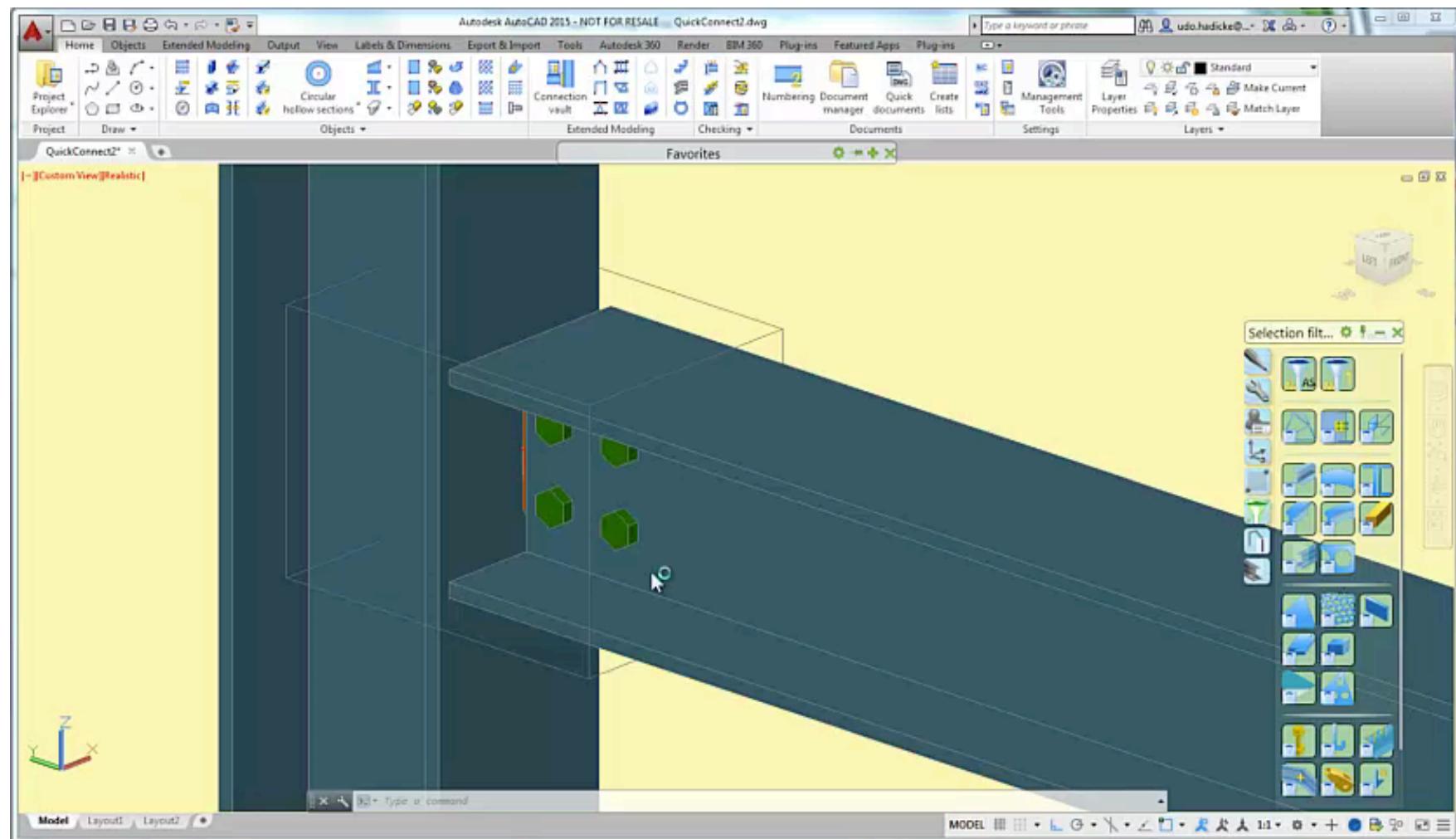
Cash receipt interface type: Printed in store (report)

Quality attributes:

Tag: ResponseTime.CashReceiptPrintLaunch ...

## Example 2: Software

### Autodesk Advance Steel + Revit Software



## Example 2: Software

### Autodesk Advance Steel + Revit Software

As a steel detailer I need to model connections so that I can determine whether the structural design will interfere with the work of other building disciplines.

Acceptance criteria

- Appearance of connection will be affected by current View level of detail
- Upon connection creation the user will be able to see the created geometry at an appropriate level of detail
- Geometry generated in Advance Steel will be dimensionally identical to geometry generated in Revit from the same parameter values.

## Example 2: Software

### Autodesk Advance Steel + Revit Software

As a steel detailer, I need to select one or more steel structural members within Revit so that I can identify which structural members should be connected.

#### Acceptance criteria

- Filter selection to structural items including both steel and concrete
- Filter selection to structural items eligible for chosen connection type
- Guided selection of eligible items
- Guided selection of items in proper order
- API access to eligible items
- Identification of connecting ends based on location



## Example 2: Software

### Autodesk Advance Steel + Revit Software

As a steel detailer, I need to select the type of connection to apply to one or more structural items so that I can begin making connection decisions based on an initial design.

#### Acceptance criteria

- Initial availability of < 20 types from Advance Steel
- Scalability to ensure eventual availability of approximately 300 types from Advance Steel
- Ability to include type choices from third parties
  - New type choices will be created outside of Revit

## Example 2: Software

### Autodesk Advance Steel + Revit Software

As a steel detailer, I need to understand the location, quantity, and diameter of the bolts at connections represented in Revit, so that I can further develop the connection detail.

#### Acceptance criteria

- User should be able to measure distances: center of bolts to side of the plates, between bolts...etc.

# Example 3: Hardware

## Wikispeed Car Suspension Module

**User Story:** "As a driver, I want to reduce suspension vibration so that I enjoy the ride more"

### Key Story Elements

- Adjust shocks to reduce stiffness of suspension module
- I/O contract between "suspension" module and "Chassis" module not impacted
- Therefore, all work can be completed within one 7-day sprint



Bolting pattern, drive train connection and hydraulic interface represent the "I/O contract"

## Example 4: Services Context and Background

- New Scrum Inc. class to teach the application of Scrum to both hardware and software
- Not sure of market interest, but determined course would be held if it attracted at least 10 students to sign up
- Decided to market on Kickstarter to test the water and go from there



**Develop eXtreme Manufacturing class and curriculum**

by Joe Justice

Create hands-on classroom experience for Agile / Scrum Design and Manufacturing, build a car with WIKISPEED at their Seattle HQ

📍 Lynnwood, WA

**Successful!**

**161% funded**   **\$16,127 pledged**   **Funded Apr 27, 2014**

# Example 4: Services

## Industry Stack and Sample User Story Vertical Slice

### “New Training” Stack



**MVP:** “Can we get 10 registrants in 60 days?”

— No “course delivery” activity needed to achieve MVP goals...comes in future stories

— Be prepared for inevitable questions and help people enroll

**KICKSTARTER**  
— Main push of MVP... get the word out

— Thought through only enough for a provocative description...main course work comes after “go” decision

— “Proof of concept” and budget-level

— Need system capable of enrolling at least 10 people

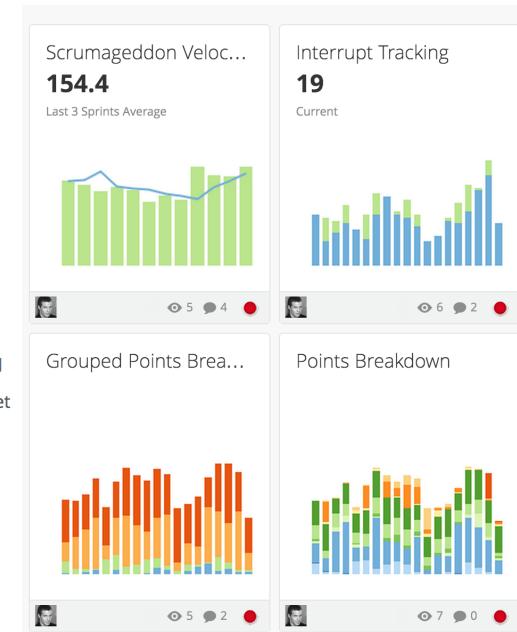
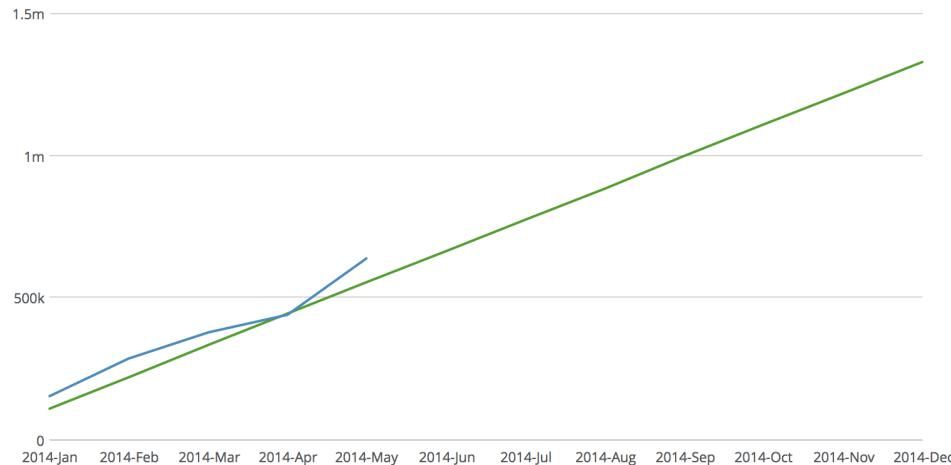
— Need to least select date and city for offering (venue nice)

# Example 5: Dashboard Context and Background

- Leadership must maintain visibility into org's progress towards vision/goals.
  - To make course adjustments as needed to ensure progress
  - Informed decisions require relevant context and metrics
- What are the right/wrong agile metrics to track?
- How do we make sure those metrics are updated with the latest data?
- How do we add and/or tweak them quickly and easily?

Net Income Burn-up by Month ▾

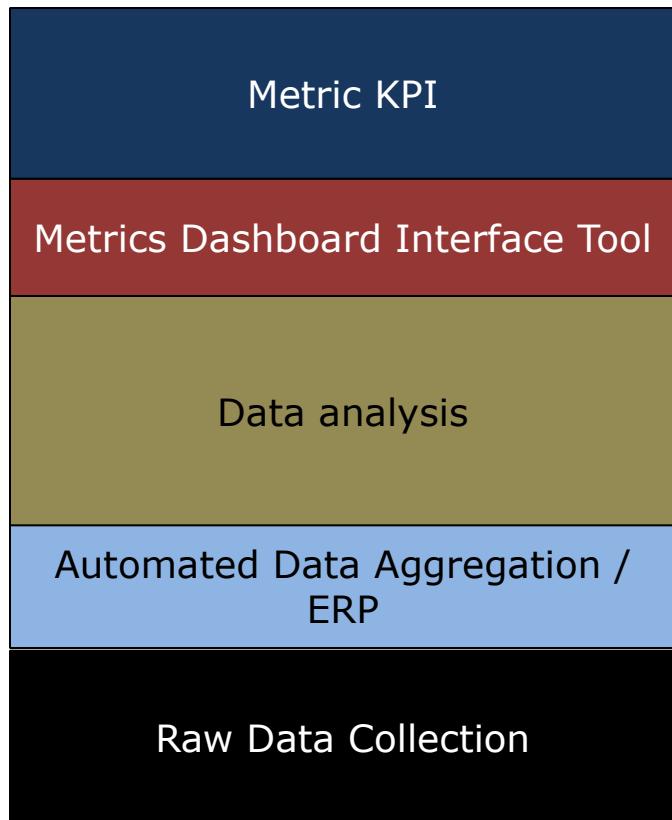
**\$638.57k** Net Income YTD



# Example 5: Dashboard

## Dashboard Stack and Sample User Story Vertical Slice

### “Dashboard” Stack



**User Story:** As leadership, we must know how net income is burning up for the year so that we can forecast profits

- Set up KPI visualization that clearly shows profit burn up
- Create location in visualization tool for the KPI and establish data update cadence
- Set up underlying analysis to calculate cumulative monthly net income data from revenues & costs
- Automate aggregation of revenue and cost data into data warehouse
- Establish method for collecting revenue and costs data for each transaction

# Conclusion

- Writing good independent user stories is vital
  - They are not held up by external dependencies
  - They can be taken rapidly all the way to Done!
  - Getting stories done in the sprint will double team velocity
- It takes both art and science to write stories well
  - The tips presented here can help get you started, but practice makes perfect
- It is as important to recognize a good user story when you see one

## Scrum and CMMI Going from Good to Great

Jakobsen, C.R. ; Sutherland, J.

Agile Conference, 2009. AGILE '09.

DOI: [10.1109/AGILE.2009.31](https://doi.org/10.1109/AGILE.2009.31)

Publication Year: 2009 , Page(s): 333 - 337

Cited by: [Papers \(8\)](#)

**IEEE CONFERENCE PUBLICATIONS**

## Teams That Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams

Sutherland, J. ; Harrison, N. ; Riddle, J.

System Sciences (HICSS), 2014 47th Hawaii International Conference on

DOI: [10.1109/HICSS.2014.580](https://doi.org/10.1109/HICSS.2014.580)

Publication Year: 2014 , Page(s): 4722 - 4728

**IEEE CONFERENCE PUBLICATIONS**

# Questions?



# Stay Connected

## E-mail

- [info@scruminc.com](mailto:info@scruminc.com)

## Twitter, Facebook, and G+

- @Scruminc, #Scrum, #Agile

## Our Website

- [www.scruminc.com](http://www.scruminc.com)
- check in for announcements, new content and services, book releases, and more!

## ScrumLab

- [Scrumlab.scruminc.com](http://Scrumlab.scruminc.com)
- Articles, videos, papers on all things scrum

## Online Courses

- Advance your learning with our interactive online courses. Visit scrumlab to view upcoming topics.