User Stories & Story Maps

Week 3

Software Product management Process

Fig. 2. SCRUM software product management process.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011, January). The agile requirements refinery: applying SCRUM principles to software product management. *Information and Software Technology, 53*(1), 58-70. doi:10.1016/j.infsof.2010.08.004

Scrum/Iterative workflow

https://www.scrum.org/resources

Introduction on how to write User Stories

There are many perspectives and purposes

https://www.youtube.com/watch?v=Pn-QMvDTuEY
FEMKE Design

Pre-sprint – User Requirements_v1

What is the "business" **problem or opportunity** (What needs to change? Why is this product worth doing?)

What is the **product goal**? (Outcome wanted). – MAKE FRONT OF MIND FOR TEAM

What do users want to be able to do to achieve the product goal?

Significant user type with characteristics that distinguish from other user types that may affect the design. NOT "User"

The new capability the user wants so they can reach smaller goal (outcome) In "business" language & NO solution details

What is the goal (desired outcome) of this new user capability?

Problem Space

User stories

As a <???> I want to be able to <???????> so that <?????????>

What criteria for the order?

Who puts them into order?

When are they put into order?

Why are they put into order?

Well thought out.
Do these next.

Placeholders for conversations.

Still a bit vague.
Do these later.

https://www.scrum.org/resources

Pre-sprint – User Requirements_v1 User Stories part 1

Significant user type with characteristics that distinguish from other user types that may affect the design. NOT "User"

The new capability the user wants so they can reach smaller goal (outcome) In "business" language & NO solution details

What is the goal (desired outcome) of this new user capability?

As a <???> I want to be able to <???????> so that <??????????>

As the PO I want to have lots of articles in the evidence repository

As a practitioner, I want lots of evidence to be available so the evidence is convincin

I want some way to be able to keep adding articles with evidence for analysis to add evi the repository

I want people to be able to add new evidence so the repo gets bigger a erage crowd sourcing

As a **researcher** I want to be able to **recommend articles to include i vidence repository** so that **the evidence available keeps expanding**

Placeholders for conversations.

We should check the article is about SE with evidence (relevance)

We should check the article is not already in the repo (avoid duplicates)

We should check the quality of the evidence in the article (quality)

The submitter should be informed/thanked if the article they submitted is accepted or not (and why not)

https://www.scrum.org/resources

Discovering Product Goal, User Needs

Knowledge about the problem, user needs and possible solutions is distributed unevenly - at the start especially.

There are many perspectives on the problem and user needs

There are many ways to design a software product to address these needs

Different world-views and language

If you just imagine the user needs from your perspective!

Documenting User Needs/Requirements - User Stories

Physical small cards

Electronic cards/tickets/table

More detailed user stories with at least one acceptance criteria

Keep breaking down until you expect that the user story will take **1-2 ideal days** and definitely **< 1 sprint**)

Work out **technical tasks** that need to be done on the next set of user stories to be worked on
Often tasks are associated with User Stories

Avoid technical "user stories" PUT IN LINK HERE

https://www.userinterviews.com/blog/how-to-write-better-user-stories-using-ux-research

USER INTERVIEWS

Specification by Example

We will come back to this

Order amount	VIP Member	Number of items in order	Free freight	Order Discount
<$100	Y	>=10	n	y
>=$100	Y	<10	y	n
<$100	Y	>=10	n	y
>=$100	Y	<10	y	n
<$100	Y	>=10	n	y
>=$100	Y	<10	y	n
<$100	N	>=10	n	n
=$100	N	<10	n	n
<$100	N	>=10	n	n
<$100	N	<10	n	n
=$100	N	>=10	n	n
<$100	N	<10	n	n

Discovering User Stories (user requirements)

Big upfront effort

plan-driven control based on high-confidence (certain), **long-term** predictions

Iterative and incremental effort

Frequent opportunity for changes based on empiricism and short learning loops and **short-term** certainty and long-term uncertainty.

User Story Workshops – product stakeholders, PO, BA, Dev, Tester etc
Stakeholders write them and group (and agree on order or in/out)?

Interviewing users or the PO – get placeholders for future conversations about needs, changes to the current situation, and some detail for user needs that are of most value to be worked on first

Observation, surveys, impact mapping, customer journey mapping……

Discovering user Stories – Mapping techniques

Mapping = **visualize** a lot of **information** showing **relationships** and some **structure**
Helps to make sense of something – and is also an information radiator

You can explore these yourself

Customer Journey Maps
Impact Mapping
Value stream mapping

"A successful customer journey map will give you real insight into what your customers want and any parts of your product, brand or process that aren't delivering."

David Weaver (co-founder of Vintage Cash Cow)

A customer journey map is a **visual** representation of the **important steps** a customer goes through to achieve a **goal with your company**.

You can get a sense of your customers'

- motivations
- needs
- pain points.

What are the characteristics of a good user story and how to write them so they meet these criteria

Independent of other user stories – minimize inter-team/inter team member dependencies that need a coordination effort

Not too much detail – this will be obtained close to the time of development of it – and stored in team members minds and maybe some acceptance criteria and some design diagrams and end up in the code functionality.

Written in the language of the user – non-technical, avoiding technical solution, can be checked/discussed by PO

Can be "tested" against acceptance criteria – otherwise we are never sure if solution meets the user story needs

We will come back to this after Sprint 1 – after we have tried writing a few user stories

Story Maps are the new Product Backlog

Story mapping is a way of structuring product features (as User Stories) to show a model of the entire product being developed

It avoids some shortcomings of a Product Backlog which just shows a prioritised list of features still be be worked on

Story Maps can be useful for:

- planning delivery product increments
- plan sprint product increments
- Support Up front product design
- Support work prioritisation
- Share and develop understanding

They provide a context for features to understand relationships between features, user types, and their purpose

Story Maps address a need…

The Problems with **Flat Backlogs**

Traditional Product Backlogs are flat; a prioritized list.

Great for answering **"what do we do next?"**

***Not** so great for:*

- Collaborative building & inspection
- Seeing how everything fits together
- Balancing a view of user-valued features with the need for iteration-size stories
- Planning coherent value-based releases

6

Risk of becoming a Feature Factory

https://www.slideshare.net/versionone/story-mapping-in-a-nutshell/6-the_problems_with_flat_backlogstraditional

Using a story map to slice out a delivery plan or sprint plan

https://www.scrumalliance.org/system/slides/118/original/christianhossa_specificationbyexamplewithgherkin.pdf?1349824954

The anatomy of a Story Map

Columns relate to user activities

Horizonal (slices) groups are priority
Importance or frequency of use

Can create other horizontal slices
Of features to represent the scope of
delivery cycles or MVP or sprint cycles

Each column has the system
features related to the user activity
for that column

https://agilevelocity.com/story-mapping-101/

User Story Mapping

By Jeff Patton

These are user types

1. User personas are laid out horizontally at the top of the map. Think of these as major categories of users that gain value from using the system.
2. Business/User goals are laid out next. These are major objectives that that the system must support, with tangible business outcomes and real business.
3. Goals are then broken up into explicit user activities. User activities are tangible, sequential events that describe what the user needs to do in order to get value.
4. Finally, activities are broken up into explicit system features. These features are laid out directly underneath the user activity that the feature realizes.

https://video.itu.dk/video/36977552/agil-systemudvikling-alistair-cockburn Background - from 15 - 20 min

http://agileconsulting.blogspot.co.nz/2012/05/integrating-story-mapping-with-kanban.html

Example

Figure 5: **The model is vertically divided into business processes.**

Figure 6: **The first system span represents the smallest set of features necessary to be minimally useful in a business context.**

http://www.jpattonassociates.com/wp-content/uploads/2015/01/how_you_slice_it.pdf

Example

https://www.scrumalliance.org/system/slides/118/original/christianhossa_specificationbyexamplewithgherkin.pdf?1349824954

Good resource

The latest articles about user story mapping

February 19, 2019 by Gergo Matyas

StoriesOnBoard staff pays a lot of attention to collect the latest, most inspiring articles from the internet. Read further to discovery fresh new materials and open the previously published collection of user story mapping content.

User Story Mapping as delivery planning

by Paddy Corry

"User Story Mapping adds a narrative element to a larger unit of value by taking a vertical backlog such as Jira or VSTS, and adding a horizontal axis across the top to represent the entire narrative of the larger unit of value. Let's call it the 'unit of value' a feature in this post. If you're not familiar with the technique, it is straight-forward, but also deceptively sophisticated." Read more...

User Story Mapping and probabilistic forecasting

by Willem-Jan Ageling

"Probabilistic Forecasting summarizes what is known about future events. You will not work with single-valued forecasts but you assign a probability to a number of different outcomes. Probabilistic Forecasting based on historical data is an alternative proposed by #NoEstimates. You don't make assumptions, you use actual data. This makes it a powerful way of

User Story Mapping Games

by Kateřina Mňuková

"User story mapping is a great technique, if everybody knows what to do, how and what to expect (more in my previous articles about story mapping). If you are worried that the workshop itself could be messy due to lack of knowledge about the user story mapping, start it with quick game. It takes maximum 30 min and it explains the purpose of story mapping more than enough and plus it's fun!" Read more...

7 Reasons to try user story mapping

by Kate Hopkins

"Typically, participants tell the story of a particular user's experience, documenting the various steps and options with sticky notes as they go. Story maps ensure that the focus stays on the user and what he or she is trying to accomplish, rather than on features or development requirements. They're an incredibly versatile tool that helps with design, prioritization, and communication." Read more...

"User Story Mapping is Big Design Upfront!"

by Willem-Jan Ageling

"Some people I know dislike User Story Mapping. They find it is Big Design Upfront (BDUF). You'd basically do upfront planning of multiple Sprints. And even if you would not do this, if you would allow new insights to be added, you waste a lot of time on the exercise that probably turns out to be largely incorrect when time passes." Read more...

https://storiesonboard.com/blog/user-story-mapping-articles

Resources by the guy who had the idea in 2005!

http://www.jpattonassociates.com/wp-content/uploads/2015/01/how_you_slice_it.pdf

https://www.jpattonassociates.com/the-new-backlog/

Article headings

FLAT BACKLOGS DON'T WORK FOR ME
THE FLAT BACKLOG IS POOR EXPLANATION OF WHAT A SYSTEM DOES
FOR A NEW SYSTEM, THE FLAT BACKLOG IS DISMAL AT HELPING ME DETERMINE IF I'VE IDENTIFIED ALL THE STORIES
RELEASE PLANNING WITH A FLAT BACKLOG IS DIFFICULT
BUILD A STORY MAP
KEEP YOUR EPICS – BUT STOP CALLING THEM THAT BECAUSE IT BOTHERS ME
WALK YOUR MAP TO TEST IT – TO GET THE BIG PICTURE
YOUR SOFTWARE HAS A BACKBONE AND A SKELETON – AND YOUR MAP SHOWS IT
KEEP YOUR MAP DISPLAYED TO COMMUNICATE THE BIG PICTURE
A DIFFERENT BACKBONE MAY BE IN ORDER FOR ADDING FEATURES TO AN EXISTING PRODUCT
IT'S A PATTERN – NOT AN INNOVATION

https://media.simplecast.com/episodes/audio/14099/TCPJeffPatton.mp3

A podcast by Jeff Patton on Talking Code

Tools …

Aha! | Blog PRODUCT USE CASES SERVICES PRICING RESOURCES COMPANY Blog Support Careers Log in Free trial Join demo

July 11, 2019

Just Launched! — New User Story Mapping Tool in Aha!

by Claire George Tweet Share 94 Share

Product and marketing teams love user story maps. How do we know? Because it is the second most popular item in our ideas portal. Many of you told us you wanted a better way to align your roadmap with the user and buyer journey. We agreed. And after quite a bit of journey mapping on our end, we are excited to introduce this important functionality so you can do just that.

> *You can now create user story maps in Aha! to align your roadmap with your customer's journey.*

CATEGORIES

Aha! Updates
New and fresh

Company Building
Grow, grow, grow

Product Management
Build products that matter

Marketing
Create breakthrough plans

Enterprise Transformation
All lasting companies change

Career Advice
Be happy

Remote Work
Be great anywhere

Earlier Approaches from FDD and Jeff De Luca!

http://www.featuredrivendevelopment.com/node/630

User Story Success Criteria

*acceptance criteria, Acceptance

A list of "rules" or criteria or tests or behaviours that should be met if the story is to be successful

Behaviour Driven Development (BDD)
Acceptance Test Driven Development (ATDD)

As a purchaser I want to be able to pay online by credit card for convenience

Possible Success criteria?

	Initial situation	Event	New situation (output)
Common template	**Given** <???????>	**When** <???????>	**Then** <???????>

https://resources.scrumalliance.org/Article/need-know-acceptance-criteria

Exercise - write down 2 user stories about SPEED

https://padlet.com/tony_clear/speed-user-stories-8paqocdeblzf3ro4

In pairs select the best 2

In teams of 4 select the best 1

A user story is written from the perspective of what the end user wants in the language of the end user – WHY?

In the Agile way of working the focus is on creating user value –user stories capture this user value and keeps the focus on it during all development

They focus on understanding the business value from the user's perspective (as explained by the PO) so that the **right thing** is built.

Delays the temptation for developers to jump to the solution before understanding the problem and the benefits of the solution.

Helps user (PO) clarify what they want

Makes it easy for the user to verify the requirement and when this will be done and everyone can share this understanding if a common language is used

In a nutshell

Skills, mind set needed
Writing user stories – what info, how much detail, what "size"
Defer detail
Asking questions to get detail
Write acceptance tests (conditions of satisfaction)
Split user stories
Forecast effort for short frequent planning
Write personas
Understand user value
Create and Use story maps
Manage changes

Short feedback cycles
Dev team learns from doing
PO learns from seeing
-> requirements emerge/change

Defer detail until the dev team need it

Act as a reminder to get the detail just in time

Split stories
Write conditions of satisfaction

User stories

Act as a reminder that dev team promises to have conversations and ask questions

Point to requirements

Conversations
Diagrams
Prototypes
User interfaces

Be sensitive to your user task's "altitude"

Too abstract

Activity or "Kite level"

Longer term goals often with no precise ending. I'll perfo
several functional tasks in the context of an activity

Think about user experience at this level

Functional or "Sea level"

I'd reasonably expect to complete this in a single sitting

Sub-Functional or "Fish level"

Small tasks that by themselves don't mean much. I'll do several of these before I reach a functional level goal

Too detailed

* from Cockburn's Writing Effective Use Cases

© Jeff Patton, all rights reserved, www.AgileProductDesign.com

Three pieces of information are useful to the development team

knowing these can influence the developers' design of the solution

As a… What is the primary actor, user role or **persona** that the requirement is for?

I want… What is the new action/feature/goal the user wants to be able to happen?

So that… What is the user's main reason for wanting the new requirement – the expected benefit?

This is just a thinking tool – you do not always have to use this format

Examples

As a user, I want to order prints of my photos

As a photographer, I want to see a preview of the photo book before I order

As a user, I want to cancel my order

As a frequent flyer, I want to rebook a past trip so that I save time booking trips I take often

Software requirements are a communication challenge

Figure 1: Design as a Mapping Process

User stories are placeholders for requirements

Analysis spans the entire lifetime of the deliverable

They are complete units that can be build tested and delivered.

User stories are worth getting right …

- ✓Set the boundaries for each work item – the unit of analysis/discussion
- ✓Are the unit for planning sprints (*estimation* for sprint backlog)
- ✓Are the unit for *keeping track* of what is still to be done overall and in what order (ordered product Backlog)
- ✓Are the unit for *testing* (acceptance tests, DoD)
- ✓Are the unit for monitoring sprint *progress* (on the story board, burndown charts)
- ✓Are the basis for *change management* (add/subtract/modify->re-order)
- ✓Are the unit for *division of labour*

DEVELOPER Task vs story

Create a database to store job seekers CVs in

As a user I want to be able to save my CV to a database

As a **job seeker** I want **my CV to be easily available to as many prospective employers as possible anytime they want** so that I **maximise my chances of being offered a job**

https://www.kununu.com/de/jobs?q=business%20analyst

Break into smaller user stories

As a **job seeker** I want **my CV to be easily available to as many prospective employers as possible anytime they want** so that I **maximise my chances of being offered a job**

WHY?

Planning, estimation
work splitting (independent)
testing

HOW?

Workflow Steps

Business rule variation eg sort

Major effort extracted - eg pay by credit card – most effort in first one

Simple vs complex – pick simple version first – American Express is harder to implement

Think about Developer tasks to create features for these

Why?

Maybe to estimate

To have the right people involved – technical specialty?

To plan development in the sprint

How much of this should I document?

Create database

Hook up to node.js

Create front end in react.js

etc

Self-organizing teams are central to the Agile way of working….

So do we need a team leader?

The team agrees on the..

- acceptance criteria (condition of satisfaction)
- proposed solution approach and
- estimate of effort

to complete each story

Creating user stories

Anyone (the team including PO and other stakeholders) can write user stories – user story workshop?

- PO is responsible for making sure a PB exists and the order
- Everyone should be involved in discussions about the user stories

They should represent value/benefit that the user wants

- I have also seen developer story cards, non-functional cards, bug cards, …..

Acceptance criteria should be written as part of the user story (*this will be done when…..*)

The initial user stories and acceptance criteria DO NOT need to be detailed or perfect or complete

Users come first

Avoid speculating imagining what users want!

If you don’t know who the users and customers are and why they would want to use the product, then you should **not** write any user stories.

Carry out the necessary user research first, for example, by observing and interviewing users.

Create stories collaboratively

https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/

Personas lead to the right stories

Ask yourself what functionality the product should provide to meet the goals of the personas

https://www.romanpichler.com/blog/personas-epics-user-stories/

How is detail added to a user story and when?

When – when it is near the top of the PB

Often teams had a reasonably clear idea of what would be in the next 2-3 sprints ahead (order and estimate).

Detail is added by

breaking stories down

Adding acceptance tests (acceptance criteria, success criteria, conditions of satisfaction)

Where are the details…

As a registered user I want to be able to cancel my order so that I can change my mind

- Does the user get a full or partial refund?
  - Is the refund to user's credit card or is it site credit?
- How far ahead must the or der be cancelled?
  - Is that the same for all orders?
  - For all customers? Different requirements by market?
- Is a confirmation provided to the user?
  - On-screen? Email? Letter?

Conditions of satisfaction as details (Acceptance criteria Success criteria, Acceptance Tests)

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests / acceptance criteria

Non-functional requirements

- For some requirements it's inefficient to be captured in user stories because they apply to all of them
  - Example 1: "The system must support peak usage of up to 50 concurrent users"
  - Example 2: "Do not make it hard to i18n the software later if needed"
- Write **constraint cards** and keep them **visible**

User stories about UI

- Keep stories with UI elements until a later sprint when stories shift from being new functionality to being modification of existing functionality
- Example: “A user can select dates from a date widget on the search screen”

Breaking user stories up

1. Workflow Steps

Identify specific steps that a user takes to accomplish a specific workflow, and then implement the workflow in incremental stages.

As a utility, I want to update and publish pricing programs to my customer.

...I can publish pricing programs to the customer's in-home display

...I can send a message to the customer's web portal.

...I can publish the pricing table to a customer's smart thermostat.

Leffingwell, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (Agile Software Development Series)* (1st ed.). Addison-Wesley Professional. p135-136

Breaking user stories up

2. Business Rule Variations

At first glance, some stories seem fairly simple. However, sometimes the business rules are more complex or extensive than the first glance revealed. In this case, it might be useful to break the story into several stories to handle the business rule complexity.

As a utility, I can sort customers by different demographics.

...sort by ZIP code.

...sort by home demographics.

...sort by energy consumption.

Breaking user stories up

4. Simple/Complex

When the team is discussing a story and the story seems to be getting larger and larger ("What about x? Have you considered y?"), stop and ask, "What's the simplest version that can possibly work?" Capture that simple version as its own story, and then break out all the variations and complexities into their own stories.

As a user, I basically want a fixed price, but I also want to be notified of critical-peak pricing events.

...respond to the time and the duration of the critical-peak pricing event.

...respond to emergency events.

Definition of Ready – meets quality criteria

(ready to be put on the PB)

The below principles of good agile stories come from the post How To Write Meaningful Agile User Stories by Isaac Sacolick

Key stakeholders must achieve a shared understanding of the deliverable

Stories should convey the opportunity, issue, need or value that it will deliver

The story title should be short and convey the deliverable without reading the details

Good stories deliver an atomic increment in business

Stories need to be completed in a Sprint

Good stories have sufficient acceptance criteria

The team should be able to estimate the story

INVEST – a quick checklist to check the usefulness (quality) of user stories

The INVEST checklist comes from INVEST in Good Stories by Bill Wake. It's an acronym with six important quality characteristics for user stories:

Independent — Can the story stand alone by itself ?

Negotiable — Can this story be changed or removed without impact to everything else?

Valuable — Does this story have value to the end user?

Estimable — Can you estimate the size of the story?

Small —Is it small enough?

Testable — Can this story be tested and verified?

INVEST

Independent
As a user, I want to pay for my order with VISA
As a user, I want to pay for my order with one type of credit card
As a user, I want to pay for my order with Mastercard
As a user, I want to pay for my order with an additional type of credit card

Negotiable – not a contract!

As a user, I want to pay for my order with my credit card

Note: Accept Visa, MasterCard and American Express. Consider Discover

As a user, I want to pay for my order with my credit card

Note: Accept Visa, MasterCard and American Express. Consider Discover. On purchases over £100, ask for card ID number from back of card. The system can tell what type of card it is from the 1st two digits of the card number. The system can store a card number for future use. Collect the expiration month and date of the card

Estimable

A job seeker can modify a CV already uploaded

- It is important for developers to be able to estimate or at least take a guess at the size of a story

3 common failures

1. Developers lack technical knowledge
2. Developers lack domain knowledge
3. Story is too big

Testable

A user must find the software easy to use
A user must never have to wait long for any screen to appear

Definition of Done

What satisfaction criteria apply to every user story so that the story is "out of my life"

May be different for different teams/sprints

IDEAS?

All unit, integration and acceptance tests are passed
Code is submitted to the repository and reviewed
The feature has been deployed on the test/staging/UAT/production environment
Documentation is completed and uploaded to the wiki
UAT is completed
CAB has signed it off

Smells…

- Too many interdependent stories -> make stories larger or slice them differently
- Goldplating -> increase visibility of tasks & workload
- Too many details -> if you run out of room, use smaller cards
- Thinking too far ahead -> stop
- Splitting too many stories
- PO can't prioritise -> close look at value

https://www.youtube.com/watch?v=0HMsh459h5c

5 Common Mistakes in User Stories

Dave Farley

Main takeaways

What are user stories?

Card

- Stories are traditionally written on note cards
- Cards may be annotated with estimates, notes, etc.

Conversation

- Details behind the story come out during conversations with the product owner

Confirmation

- Acceptance tests confirm whether the story was coded correctly

Questions and Comments….