



CSSE3012 The Software Process

User Stories



Our Agenda

- Stakeholders
- Scope
- Story Identification Activities
- What is a Story
- Finding Stories
- Writing Good Stories



Stakeholders



<http://buffy.wikia.com/wiki/Stake>

Stakeholders

- Interested in project / system outcomes
 - Sponsor
 - and subordinates
 - End-users
 - Interested parties
 - Related project teams
- Interested in project conduct
 - Process improvement team
 - QA team



Our Agenda

- Stakeholders
- Scope
- Story Identification Activities
- What is a Story
- Finding Stories
- Writing Good Stories



Scope

- Goals ÷ (Resources × Time)
 - what's important in a feasible time period
- Decide what's
 - in scope
 - out of scope
- Informed by BMC
- Assumptions upon which decisions are based



Our Agenda

- Stakeholders
- Scope
- Story Identification Activities
- What is a Story
- Finding Stories
- Writing Good Stories



Story Identification

GOAL: Create a list of stories that need to be developed in order for the application to be delivered.



Story Identification Activities

- Conducted collaboratively between customer representatives and developers
- Create a list of stories
 - describes one thing the system should do
- Add high-level acceptance criteria (if known)
 - can be added later
 - need to become tests



Our Agenda

- Stakeholders
- Scope
- Story Identification Activities
- **What is a Story**
- Finding Stories
- Writing Good Stories



What is a Story?

- Short description of functionality
 - textually small
 - short development time (1-3 days)
- From the user's perspective
- Valuable to the user or sponsor
 - consider both types of clients
- Must be testable
- Provides enough information to make rough estimates



Why Stories?

- History of poor requirements capture
 - large out-dated documents
- Communication
 - track requirements
 - cards & BVC
 - up-to-date
 - conversation
 - understand user needs
 - conversation & confirmation

Story Structure

- As a [role], I want to [do / see / change something] so that [outcome].

e.g.

- As a *permanent employee* I want to be able to *see my leave balance* so that *I can plan my holidays*.



Story Tools

- Commonly written on index cards
 - then stuck on walls
- Can be managed electronically
 - Jira Agile
 - <https://www.atlassian.com/software/jira>
 - GitHub / GitLab Issues
 - GitHub Project
 - GitLab Issue Board
 - Trello

Story Cards

Front of Card

IB

As a student I want to purchase
a parking pass so that I can
drive to school

Priority: ~~1000~~ Should
Estimable: 4

Back of Card

Confirmations:

~~The student must pay the correct amount~~
One pass for one month is issued at a time
The student will not receive a pass if the payment
isn't sufficient
The person buying the pass must be a currently
enrolled student.

The student may only buy one pass per month.

Story Wall



Story Example

Story Ref #	Feature	Story Title	As a	I want to	so that
1	Patient Tracking	Virtual Fence	Concerned Carer	Know when my patient leaves a defined area	I can check to see if they are lost

Questions	Comments	Business Priority	Story Point Estimate	Status	Release	Iteration
Where is virtual fence defined? Which component raises alarm?	Use Google Maps	High	4	UAT	1	2

The Whole Story

- *Card* – initial written description
 - often on index cards
- *Conversation* – between developers and customer representatives
 - customer driven design
- *Confirmation* – tests to determine when implementation complete
 - initial criteria written on story card
 - full tests in automated test suite



Conversations

- Details are discovered by talking with the customer representatives
- Conversations occur whenever someone needs information
 - not “once off”
- Story cards are a starting point for a conversation
 - they don’t record the requirements
 - may record notes on card during conversation



Confirmation – Story Acceptance Criteria

- Describes customer expectations of how system will deliver story functionality
- Initially recorded as what should happen
 - simple criteria
 - typically on back of story card
- Turned into user acceptance tests (UATs) during development
 - ideally automated tests
- UATs become the requirements
 - we're done when we pass the tests



Acceptance Criteria Example

As a Potential Customer I want the system to tell me how much solar power I might generate with the selected equipment so that I can compare this to my current power usage

Acceptance Criteria

- Output displays average solar power generation for:
 - one year
 - each seasonal quarter
 - daily
- Output displays peak daily power generation

Acceptance Criteria Example

As a student I want to purchase a parking pass so that I can attend uni

Acceptance Criteria

- One pass, for one month, is issued at a time
- Student will not receive a pass if the payment isn't sufficient
- Person purchasing a pass must currently be enrolled as a student
- Student may only buy one pass per month

INVEST in your Stories

- Independent
 - dependencies make planning, prioritisation and estimation difficult
- Negotiable
 - details are worked out in conversation
 - between developers and customers
 - too much detail limits the conversation and options
 - too easy to think all detail is in story
- Valuable
 - must provide value to customer
 - get customer to write stories



INVEST in your Stories

- Estimable
 - at least to start with ballpark estimates
 - prioritisation and planning depends on this
 - problems: lack of domain knowledge or story too big
- Small
 - representing a few days in person effort
 - the smaller the stories, the more accurate the estimates
- Testable
 - need completion criteria
 - we don't develop what we can't test



Example Stories

- As a *lecturer* I want to be able to *see a list of all students enrolled in my classes* so that *I can see class lists and numbers enrolled*.
- As a *coursework student* I want to be able to *see a list of all available offerings of my classes from which I can select classes to attend* so that *I can choose convenient times to be on campus*.
- As a *course coordinator* I want to be able to *update a course profile* so that *it is up-to-date and accurate*.

Example Stories

- As a *marketer* I want *the international student portal to look like a UQ site* so that *we project a consistent image*.
- As an *IT support* I want to be able to *post an outage / downtime notification* so that *I can inform people that the portal is out of service*.
- As a *web developer* I want to be able to *capture usage data in XML log files* so that *we can analyse patterns of usage*.

Our Agenda

- Stakeholders
- Scope
- Story Identification Activities
- What is a Story
- Finding Stories
- Writing Good Stories



“Trawling” for Stories

- Story writing workshops
- Written by users
- User interviews
- Questionnaires
- Observation
- Use the information currently available
 - function / feature list
 - solution definition



Story Writing Workshop

- Include those who can contribute to the stories
 - developers, customer reps, users, ...
 - keep team small
- 1 – 2 hour session
 - to keep everyone focussed
- Brainstorm list of stories
 - filter them later
- Maintain a list of issues
 - deal with after the workshop

Workshop Tips

- Roles – Identify types of users
 - be specific
- Goals – For each role
 - general expectations of system functionality
- Depth First
 - focus on one aspect of the system at a time
 - like stories breed like rabbits
- Clarify workflows
 - use whiteboards or post-it-notes

Story Brainstorming

- Choose one aspect of system
- Everyone starts writing stories on cards
 - ~10 minutes
 - stop when people start slowing down
 - avoids facilitator filtering
- Review stories
 - 3 stacks
 - keep
 - fix
 - throw away

Reviewing Stories

- After each brainstorming session
 - each author reads out their stories
- Keep Stack
 - clear, in scope, meets INVEST principles
- Fix Stack
 - do not meet INVEST principles
 - in scope but not clear, too large, too small, ...
- Throw Away Stack
 - duplicates, out of scope
 - do not provide value to user or customer
 - e.g. focuses on technical issues

User Interviews

- Real users are best
 - proxies can work if real users not available
- Interview users representing each role
- Use open-ended questions
 - don't predetermine the answer by the question
- Developers can provide ideas
 - don't assume the users know the best solution

Our Agenda

- Stakeholders
- Scope
- Story Identification Activities
- What is a Story
- Finding Stories
- Writing Good Stories



Story Writing Guidelines

- Start with goals
 - for each role identify the goals they have for using the system
- Write closed stories
 - allow the user to accomplish something useful
- Ignore the UI
- Write for one specific user
- Use active voice
- Focus on the next few iterations
 - small estimable stories
 - larger more general stories for more distant future

Epics

- Big, general story
 - higher-level view of what the user wants to accomplish
 - too big to implement in a reasonable time
 - too big to estimate
- Useful placeholder for work to do later
 - flesh out detail when needed



Epics to Stories

- Split into small stories
 - each story needs to provide a meaningful benefit
- Provide end-to-end functionality
 - “slicing the cake” – Bill Wake



Epic Example

- As a hotel operator, I want to set the optimal rate for rooms in my hotel.
- Stories
 - As a hotel operator, I want to set the optimal rate for rooms based on prior year pricing.
 - As a hotel operator, I want to set the optimal rate for rooms based on what hotels comparable to mine are charging.
 - still an epic
 - ...

Epic Example (cont.)

- As a hotel operator, I want to set the optimal rate for rooms based on what hotels comparable to mine are charging.
- Stories
 - As a hotel operator, I can define a comparable set of hotels.
 - As a hotel operator, I can add a hotel to a comparable set.
 - As a hotel operator, I can remove a hotel from a comparable set.
 - As a hotel operator, rates charged at hotels in a comparable set are used to determine rates at my hotel.
 - ...

Goal ≠ Story

- What the system needs to accomplish or support
- Larger and more complicated than a story
 - usually not estimable, small or testable
- Break down into smaller pieces
- Good prompt for story discovery

Activity ≠ Story

- What users can do with the application
- Not valuable in and of themselves
- Usually part of another story
 - may need to create new story to support activity

Task ≠ Story

- Things the development team need to do
- No business value by themselves
- Usually merge with another story
 - may need to create new story to include task

Non-Functional Requirements

- Constraints on system behaviour
 - criteria to judge system effectiveness
- “ilities”
 - stability
 - reliability
 - usability
 - portability
 - **scalability**
 - **Maintainability**
 - **efficiency**
 - ...
- Need to be understood & captured
 - agile principles: communication and flexibility

Non-Functional Requirements

- Constraints – written like stories?
 - As a *call centre operator* I want the system to *retrieve data in less than 1 second* so that *I can respond to customer queries with no delays*.
- Infrastructure stories
 - As a *web admin* I want *a web server set up by Dec. 1* so that *we can launch external beta testing*.
- Stories don't work for everything
 - e.g. document an API as a contract

Reading

- <https://www.mountaingoatsoftware.com/agile/user-stories>
- <https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/>
- User Story Template Explanation on BlackBoard
- User Stories Examples on BlackBoard
 - complete set of stories for developing Scrum Alliance website

Next Steps

- Lecture
 - Prioritisation
 - Estimation
- Tutorials
 - User Story Identification

