Week:
2 of 12

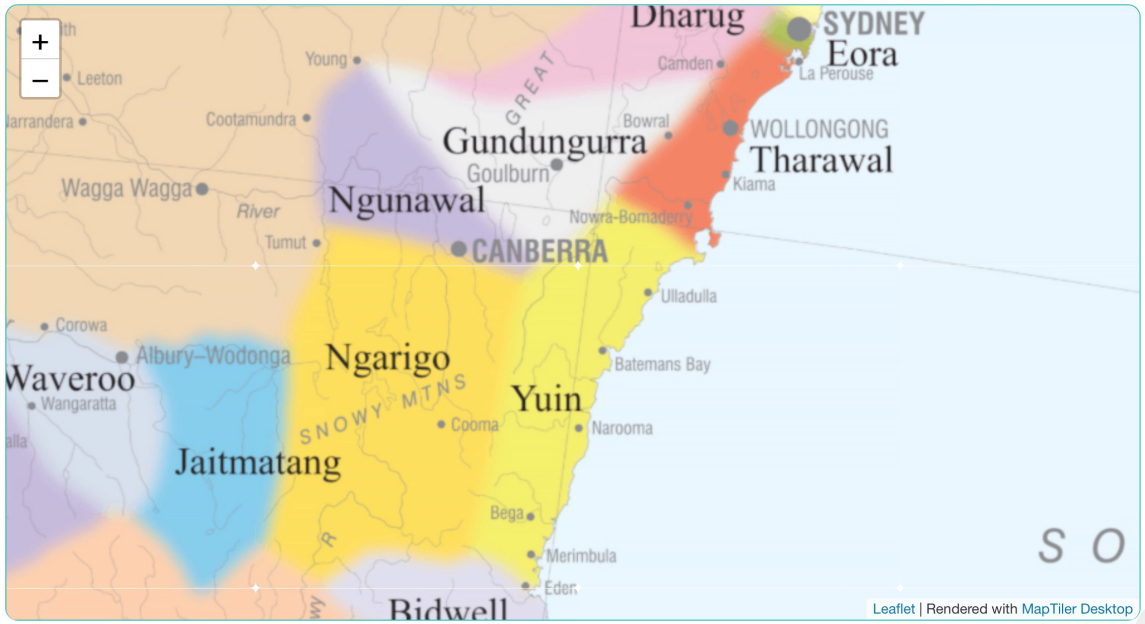
COMP 2120 / COMP 6120

AGILE (CONTINUED)

A/Prof Alex Potanin

ANU Acknowledgment of Country

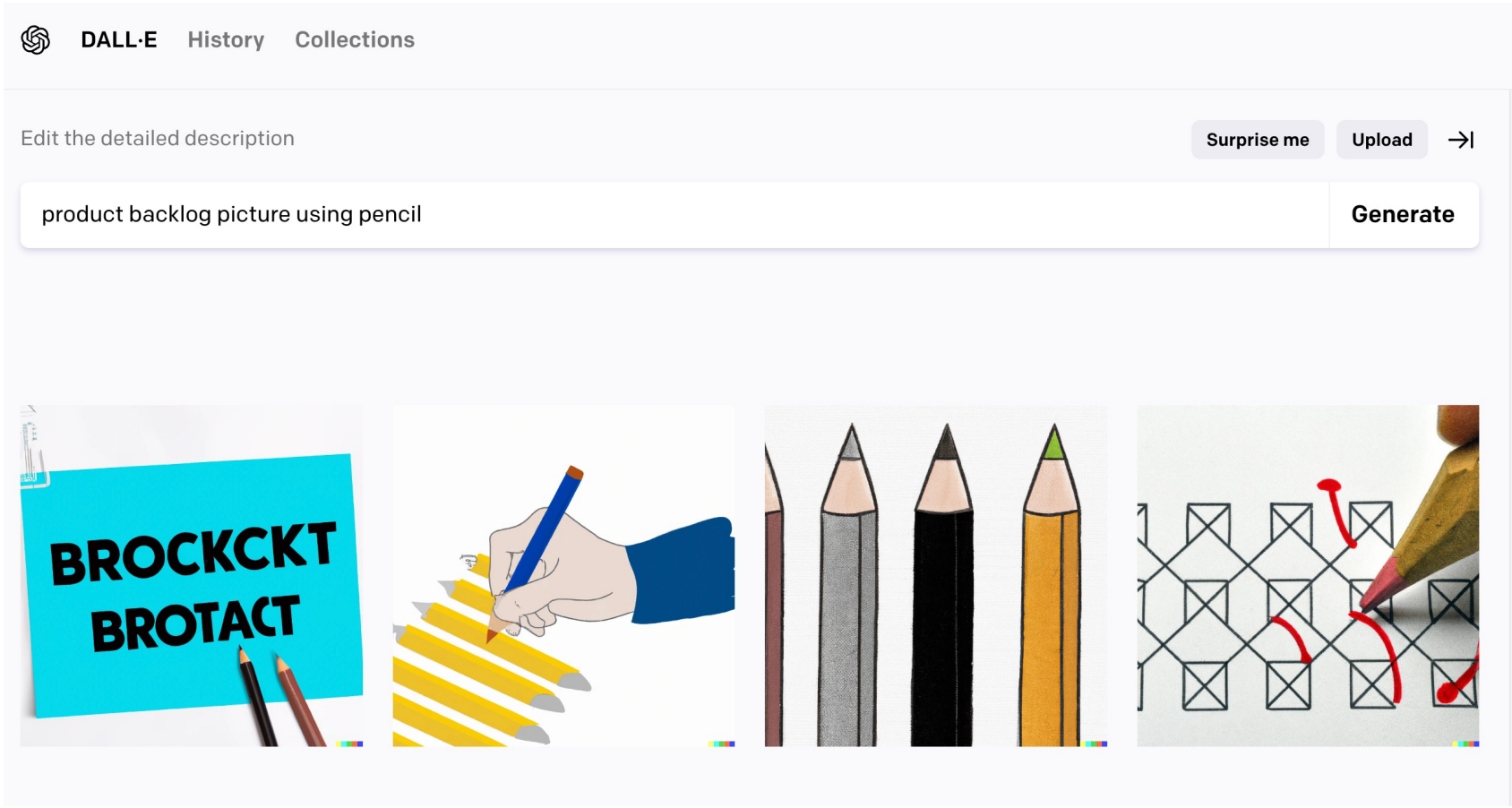
"We acknowledge and celebrate the First Australians on whose traditional lands we meet, and pay our respect to the elders past and present."



https://aiatsis.gov.au/explore/map-indigenous-australia

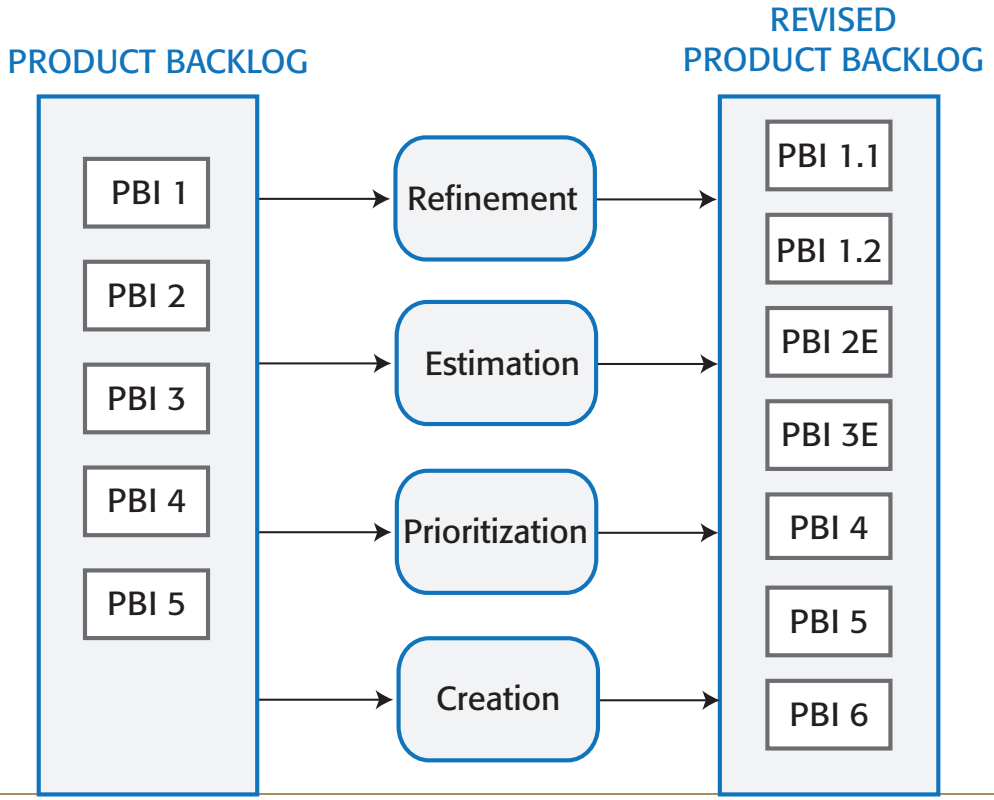
Today

- Product Backlog
- User Stories
- Sprints
- Personas
- Scenarios
- Features



Product Backlog

Product Backlog Activities



Product Backlog Activities

Refinement

Existing PBIs are analysed and refined to create more detailed PBIs. This may lead to the creation of new product backlog items.

Estimation

The team estimate the amount of work required to implement a PBI and add this assessment to each analysed PBI.

Product Backlog Activities

Creation

New items are added to the backlog. These may be new features suggested by the product manager, required feature changes, engineering improvements, or process activities such as the assessment of development tools that might be used.

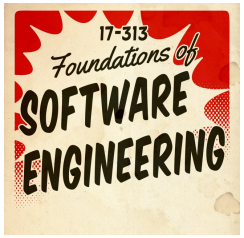
Prioritization

The product backlog items are reordered to take new information and changed circumstances into account.

PBI Estimation Metrics

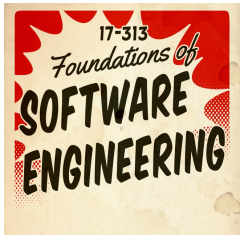
- Effort required
  - This may be expressed in person-hours or person-days i.e. the number of hours or days it would take one person to implement that PBI. This is not the same as calendar time. Several people may work on an item, which may shorten the calendar time required.
- Story points
  - Story points are an arbitrary estimate of the effort involved in implementing a PBI, taking into account the size of the task, its complexity, the technology that may be required and the 'unknown' characteristics of the work.
  - They were derived originally by comparing user stories, but they can be used for estimating any kind of PBI.
  - Story points are estimated relatively. The team agree on the story points for a baseline task and other tasks are estimated by comparison with this e.g. more/less complex, larger/smaller etc.

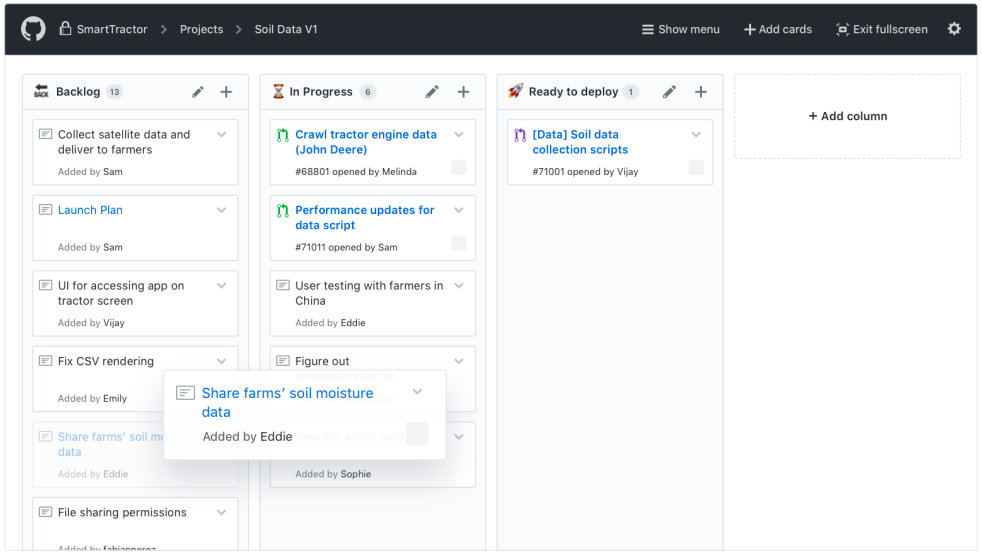
Product Backlog/Sprint Backlog



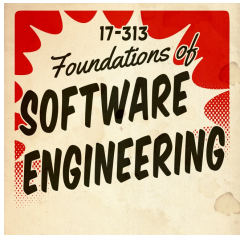
- The product backlog is all the features for the product
- The sprint backlog is all the features that will be worked on for that sprint. These should be broken down into discrete tasks:
  - Fine-grained
  - Estimated
  - Assigned to individual team members
  - Acceptance criteria should be defined
- User Stories are often used

Backlog – information radiators



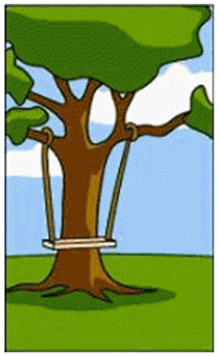


Scrum Meetings

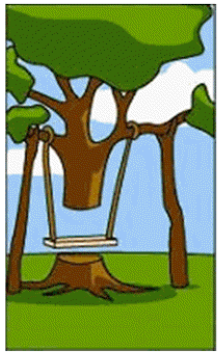


- Sprint Planning Meeting
  - Entire Team decides together what to tackle for that sprint
- Daily Scrum Meeting
  - Quick Meeting to touch base on :
    - What have I done? What am I doing next? What am I stuck on/need help?
- Sprint Retrospective
  - Review sprint process
- Sprint Review Meeting
  - Review Product

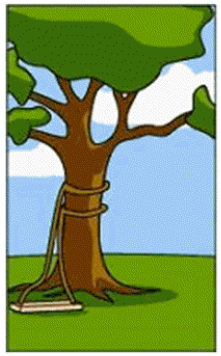
How the customer explained it



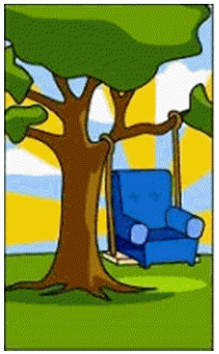
How the project leader understood it



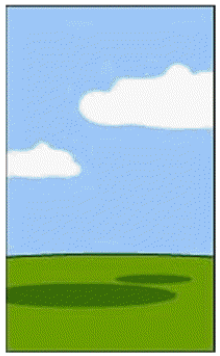
How the engineer designed it



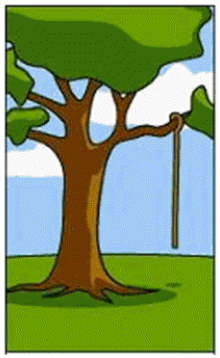
How the programmer wrote it



How the sales executive described it

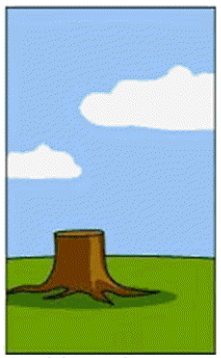


How the project was documented

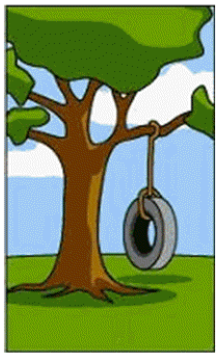


What operations installed

How the customer was billed

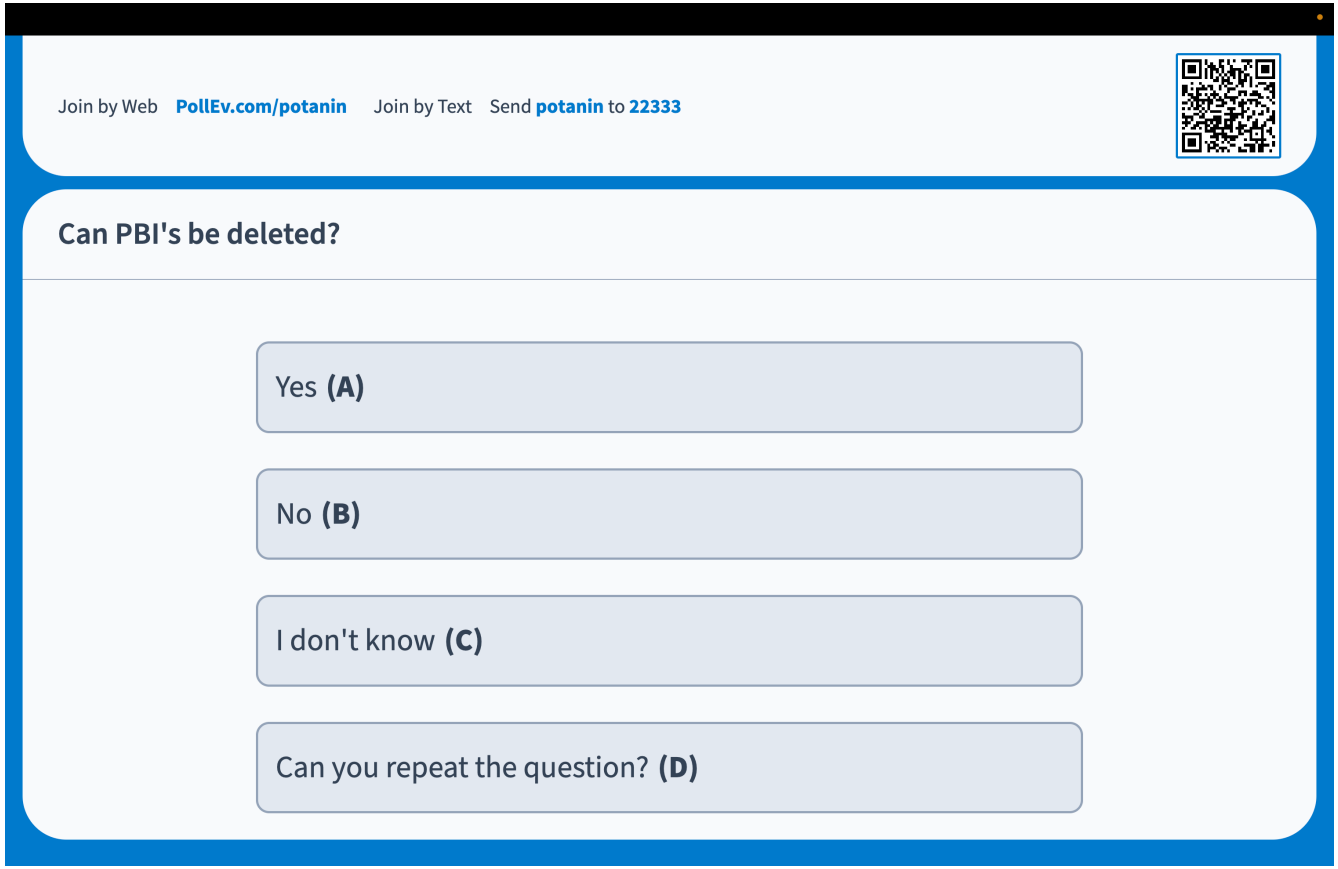


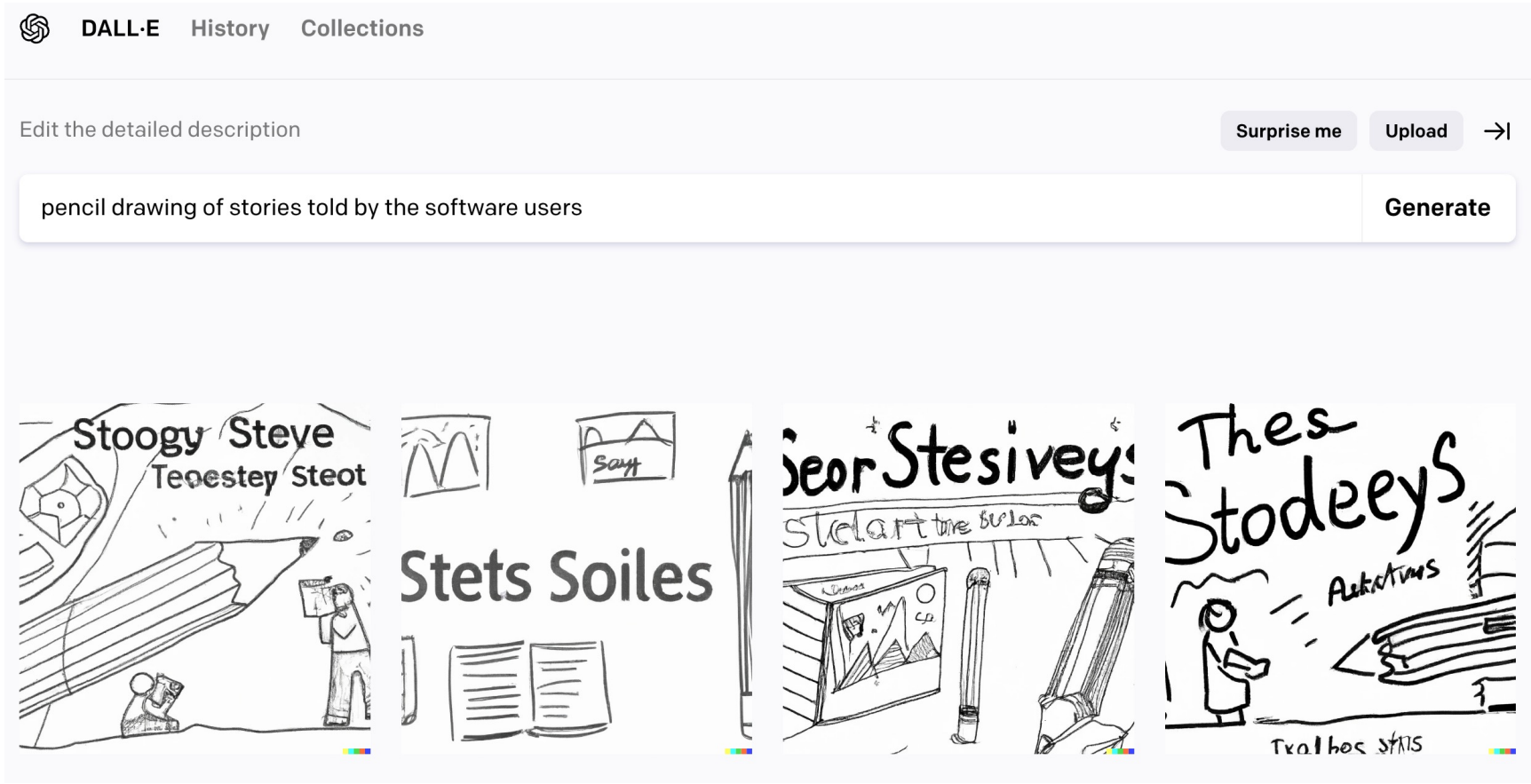
How the help desk supported it



What the customer really needed

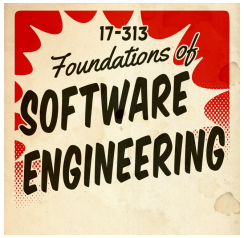
Poll Everywhere Time!

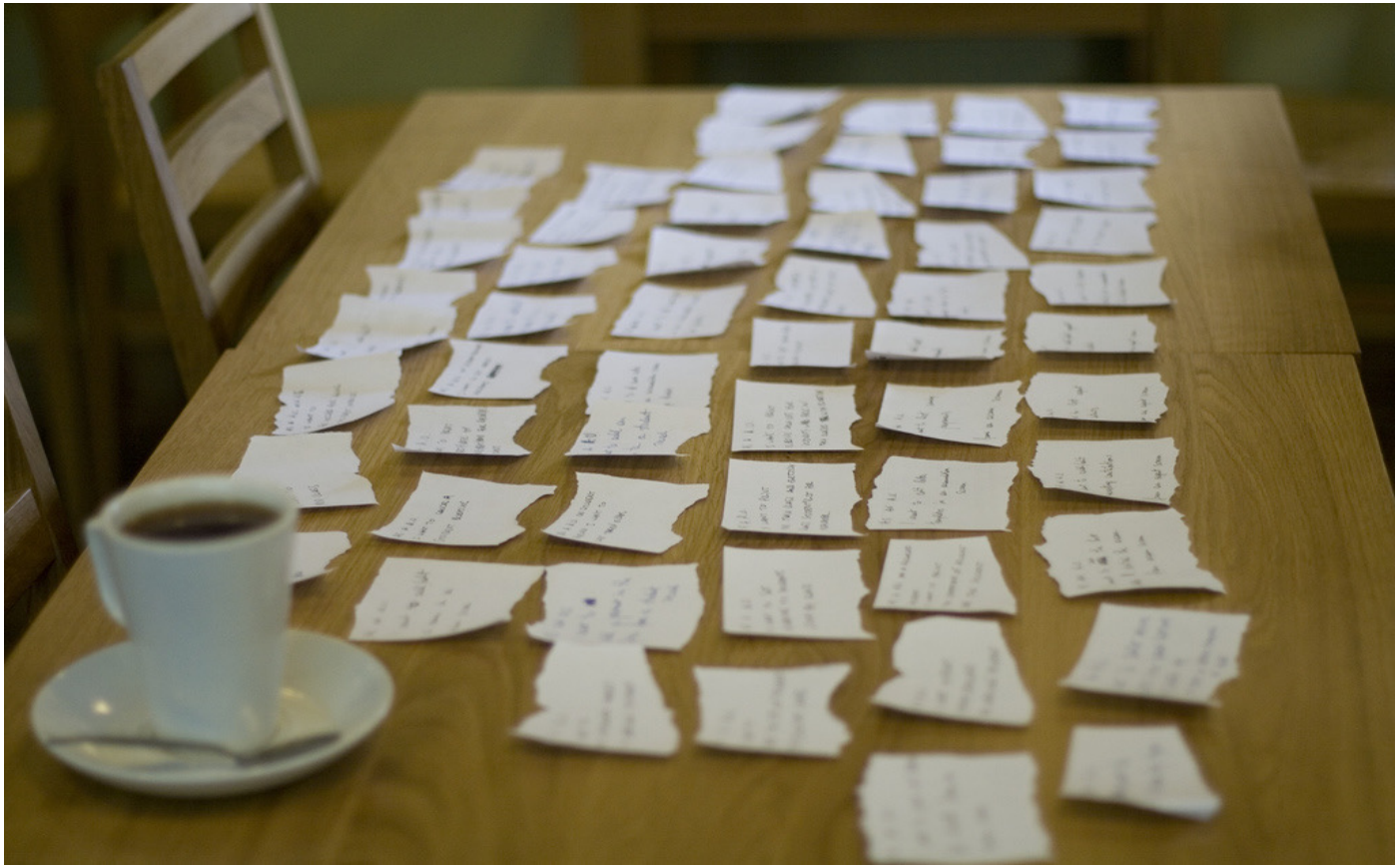




User Stories

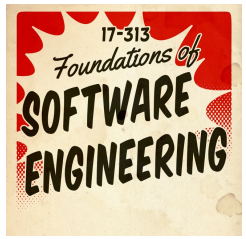
User Stories

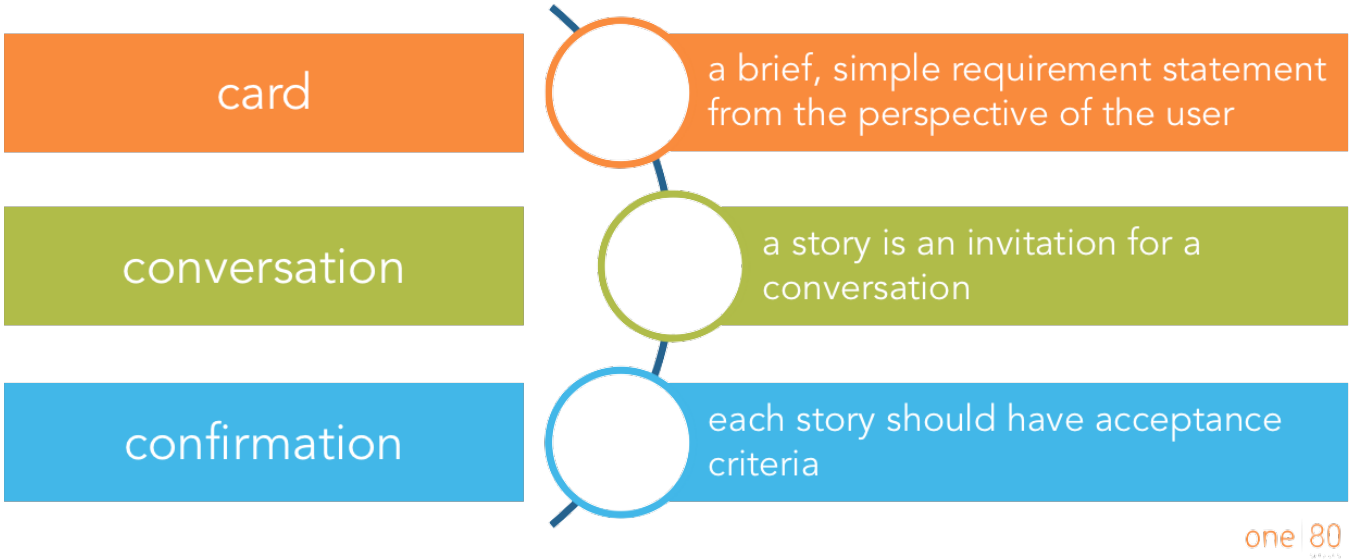




Source: https://www.flickr.com/photos/jakuza/2728096478

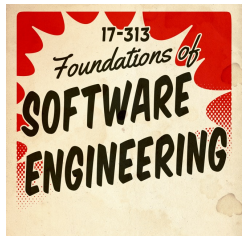
User Stories





Source: http://one80services.com/user-stories/writing-good-user-stories-think-its-not-about-writing/

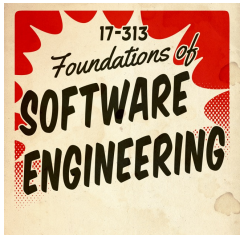
The card: user story template



"As a [role], I want [function], so that [value]"

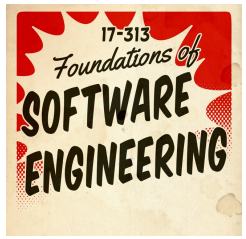
(Should fit on a 3x5 card)

The conversation



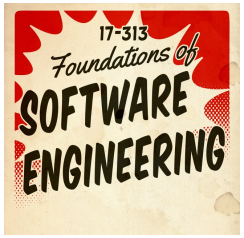
- An open dialog between everyone working on the project and the client
- Split up Epic Stories if needed

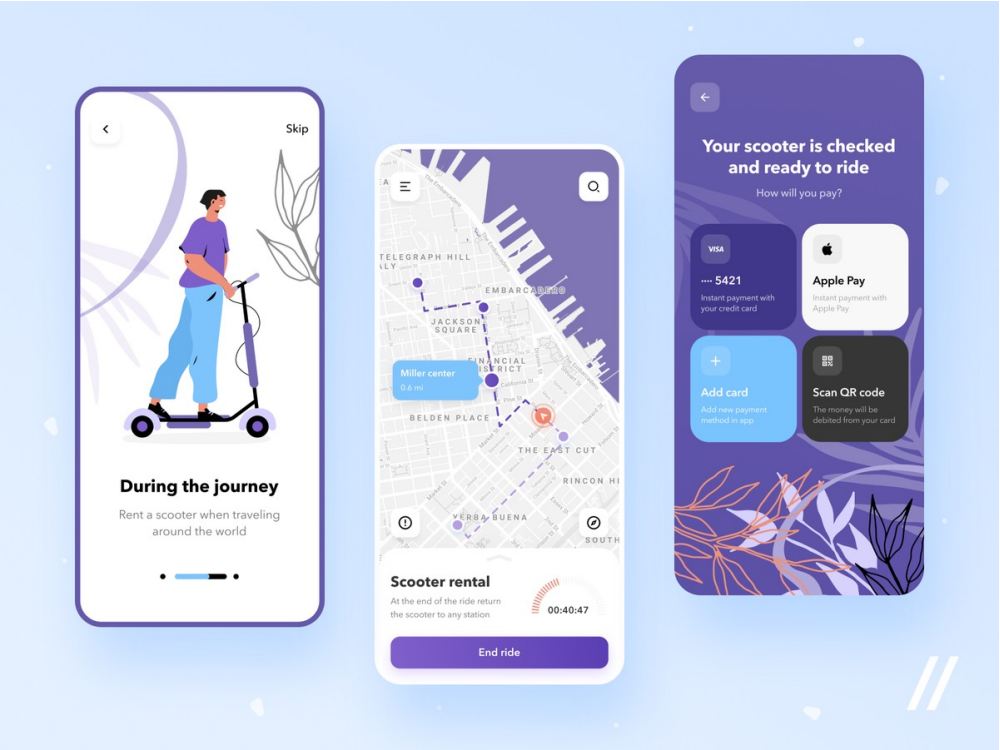
The Confirmation



- A confirmation criterion that will show when the task is completed
- Could be automated or manual

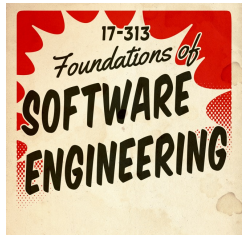
Write a User Story but Keep It Till Later





https://dribbble.com/shots/12512417-Scooter-Rental-App-Design

How to evaluate user story?



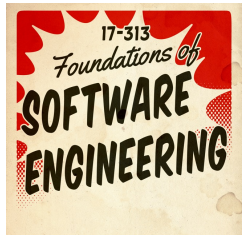
Follow the INVEST guidelines for good user stories!

Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/

one 80

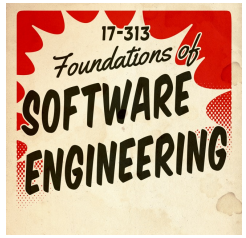
Independent

- Schedule in any order.
- Not overlapping in concept
- Not always possible



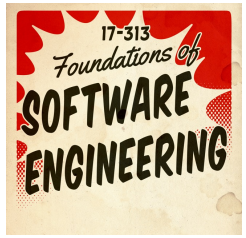
Negotiable

- Details to be negotiated during development
- Good Story captures the essence, not the details

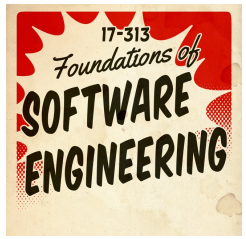


Valuable

- This story needs to have value to someone (hopefully the customer)
- Especially relevant to splitting up issues



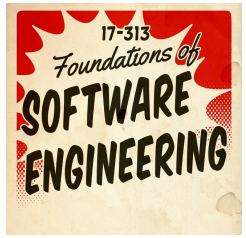
Estimable



- Helps keep the size small
- Ensure we negotiated correctly
- “Plans are nothing, planning is everything” -Dwight D. Eisenhower

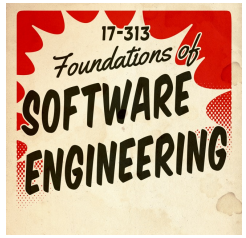
Small

- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate

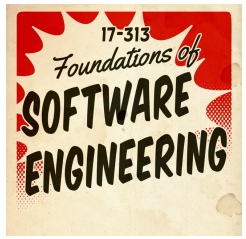


Testable

- Ensures understanding of task
- We know when we can mark task “Done”
- Unable to test == do not understand



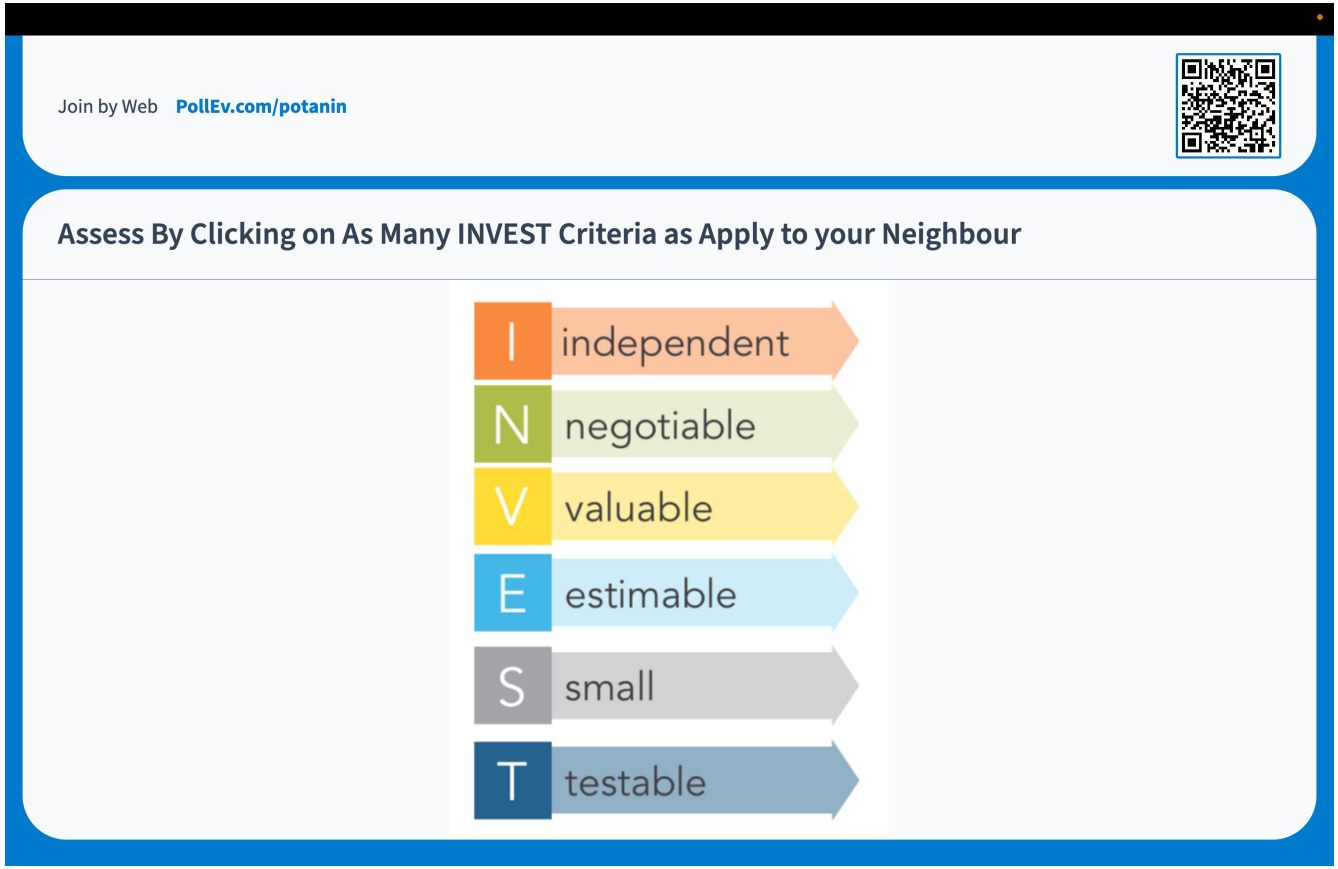
Exercise (Continued – Assess Your Neighbour's)

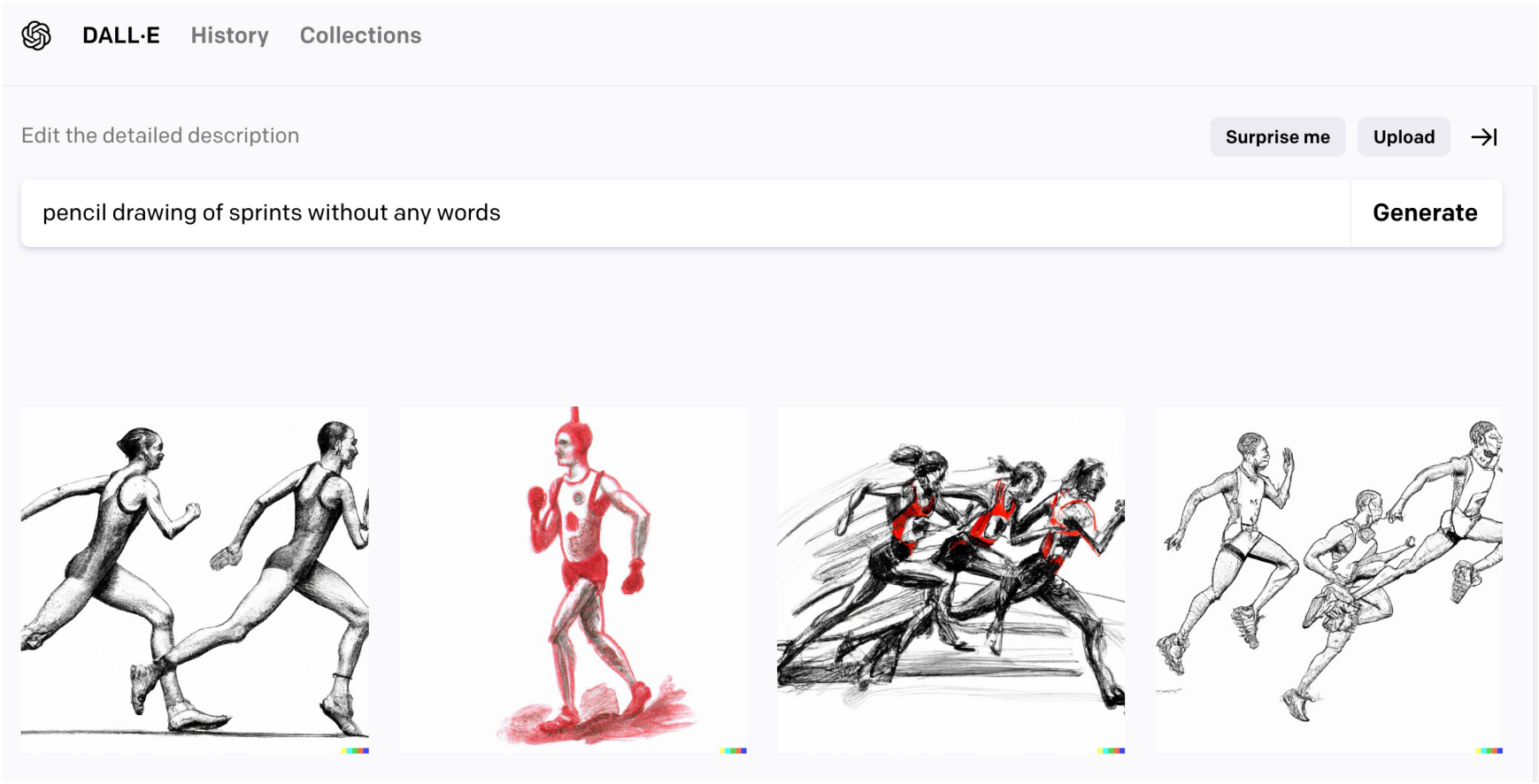


Follow the INVEST guidelines for good user stories!

one 80

Poll Everywhere Time!




DALL·E
History
Collections
Edit the detailed description
Surprise me
Upload
pencil drawing of sprints without any words
Generate

Sprints

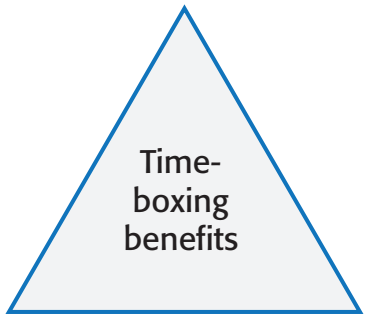
Timeboxed Sprints

- Products are developed in a series of sprints, each of which delivers an increment of the product or supporting software.
- Sprints are short duration activities (1-4 weeks) and take place between a defined start and end date. Sprints are timeboxed, which means that development stops at the end of a sprint whether or not the work has been completed.
- During a sprint, the team work on the items from the product backlog.

Benefits of Using Timeboxed Sprints

There is a tangible output (usually a software demonstrator) that can be delivered at the end of every sprint.

Demonstrable progress



Problem discovery

If errors and omissions are discovered the rework required is limited to the duration of a sprint.

Work planning

The team develops an understanding of how much work they can do in a fixed time period.

Sprint Activities

Sprint planning

Work items to be completed in that sprint are selected and, if necessary, refined to create a sprint backlog. This should not last more than a day at the beginning of the sprint.

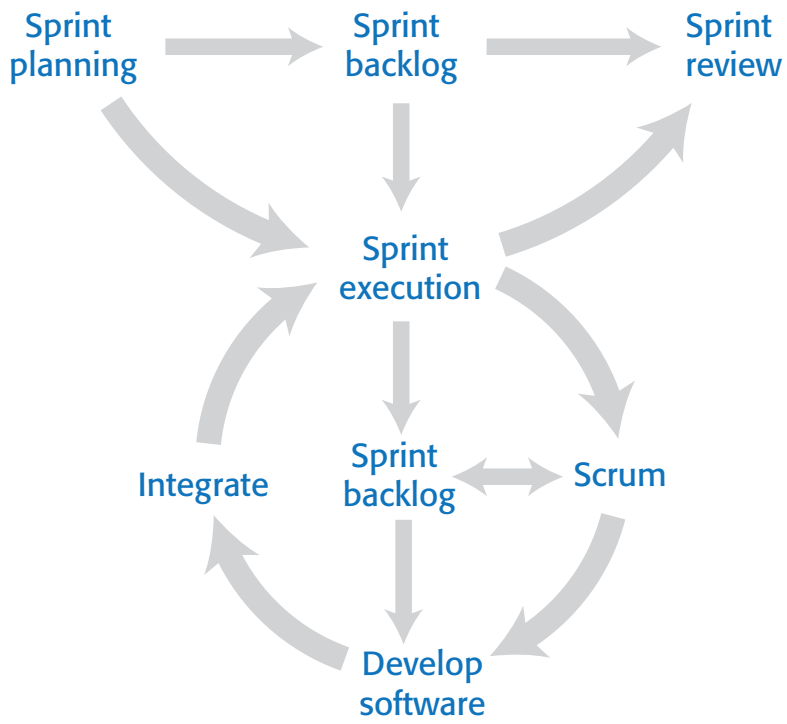
Sprint execution

The team work to implement the sprint backlog items that have been chosen for that sprint. If it is impossible to complete all of the sprint backlog items, the sprint is not extended. The unfinished items are returned to the product backlog and queued for a future sprint.

Sprint reviewing

The work done in the sprint is reviewed by the team and (possibly) external stakeholders. The team reflect on what went well and what went wrong during the sprint with a view to improving their work process.

Sprint Activities



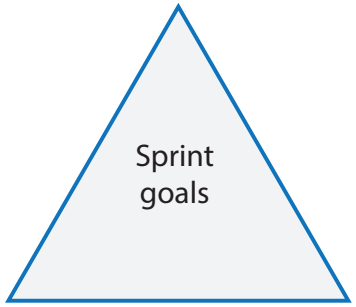
Sprint Planning

- Establish an agreed sprint goal
  - Sprint goals may be focused on software functionality, support or performance and reliability
- Decide on the list of items from the product backlog that should be implemented
- Create a sprint backlog.
  - This is a more detailed version of the product backlog that records the work to be done during the sprint

Sprint Goals

Implement user roles so that a user can select their role when they login to the system

Functional



Support

Develop analytics that maintain information about the time users spend using each feature of the system.

Performance and reliability

Ensure that the login response time is less than 10 seconds for all users where there are up to 2000 simultaneous login connections.

Sprint Planning

- In a sprint plan, the team decides which items in the product backlog should be implemented during that sprint.
  - Key inputs are the effort estimates associated with PBIs and the team's velocity
- The output of the sprint planning process is a sprint backlog.
  - The sprint backlog is a breakdown of PBIs to show the what is involved in implementing the PBIs chosen for that sprint.
- During a sprint, the team have daily meetings (scrums) to coordinate their work.

Scrums

- A scrum is a short, daily meeting that is usually held at the beginning of the day. During a scrum, all team members share information, describe their progress since the previous day's scrum, problems that have arisen and plans for the coming day. This means that everyone on the team knows what is going on and, if problems arise, can re-plan short-term work to cope with them.
- Scrum meetings should be short and focused. To dissuade team members from getting involved in long discussions, they are sometimes organized as 'stand-up' meetings where there are no chairs in the meeting room.
- During a scrum, the sprint backlog is reviewed. Completed items are removed from it. New items may be added to the backlog as new information emerges. The team then decide who should work on sprint backlog items that day.

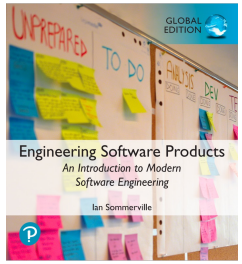
Agile Activities

Scrum does not suggest the technical agile activities that should be used. However, I think there are two practices that should always be used in a sprint.

Test automation
As far as possible, product testing should be automated. You should develop a suite of executable tests that can be run at any time.

Continuous integration
Whenever anyone makes changes to the software components they are developing, these components should be immediately integrated with other components to create a system. This system should then be tested to check for unanticipated component interaction problems.

Code Completeness Checklist



Reviewed

The code has been reviewed by another team member who has checked that it meets agreed coding standards, is understandable, includes appropriate comments, and has been refactored if necessary.

Unit tested

All unit tests have been run automatically and all tests have executed successfully.

Code Completeness Checklist

Integrated

The code has been integrated with the project codebase and no integration errors have been reported.

Integration tested

All integration tests have been run automatically and all tests have executed successfully.

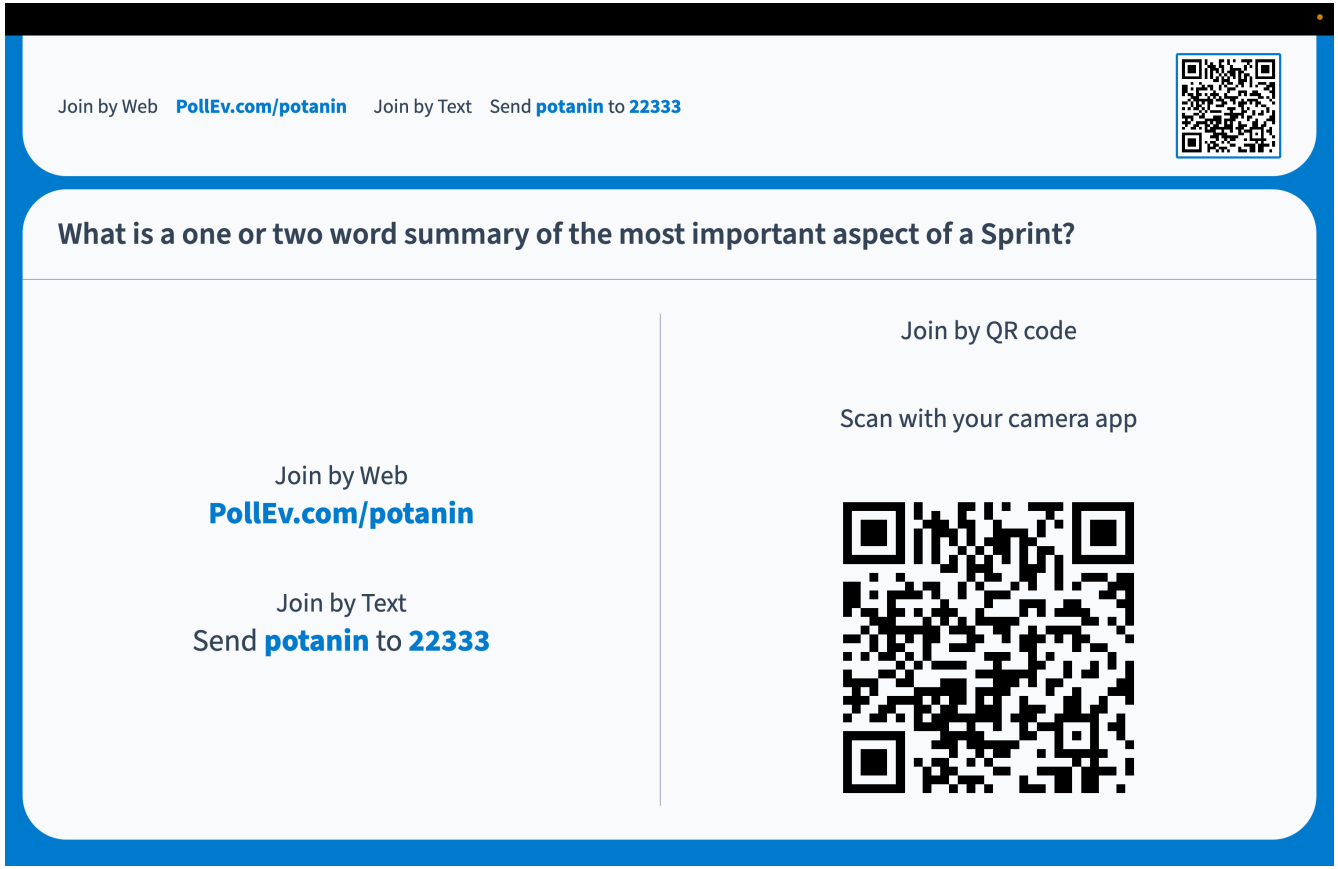
Accepted

Acceptance tests have been run if appropriate and the product owner or the development team have confirmed that the product backlog item has been completed.

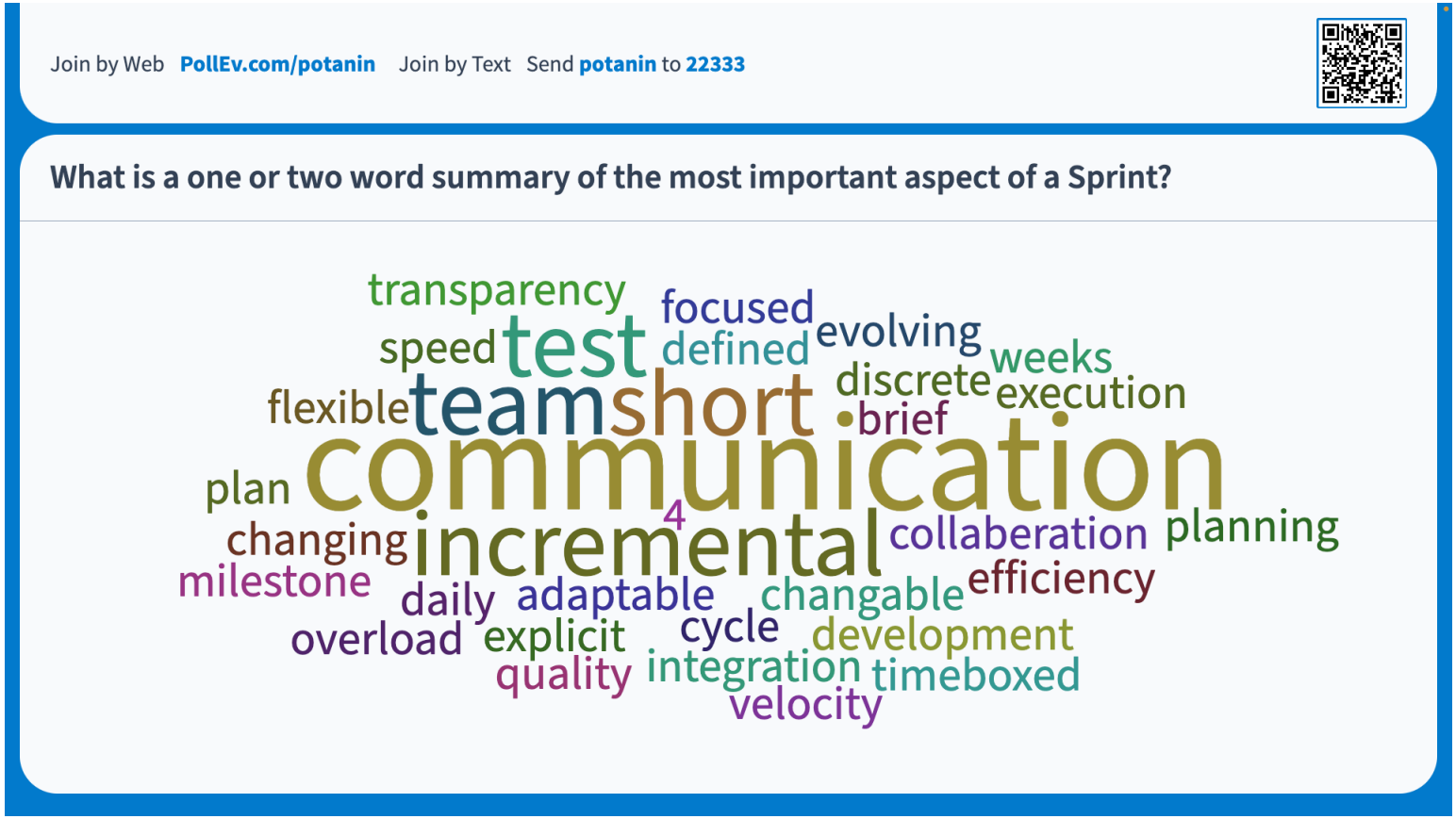
Sprint Reviews

- At the end of each sprint, there is a review meeting, which involves the whole team. This meeting:
  - reviews whether or not the sprint has met its goal.
  - sets out any new problems and issues that have emerged during the sprint.
  - is a way for a team to reflect on how they can improve the way they work.
- The product owner has the ultimate authority to decide whether or not the goal of the sprint has been achieved. They should confirm that the implementation of the selected product backlog items is complete.
- The sprint review should include a process review, in which the team reflects on its own way of working and how Scrum has been used.
  - The aim is to identify ways to improve and to discuss how to use Scrum more productively.

Poll Everywhere Time!



Poll Everywhere Time!



DALL·E History Collections

Edit the detailed description

Surprise me Upload

pencil drawing of an important persona without any words

Generate

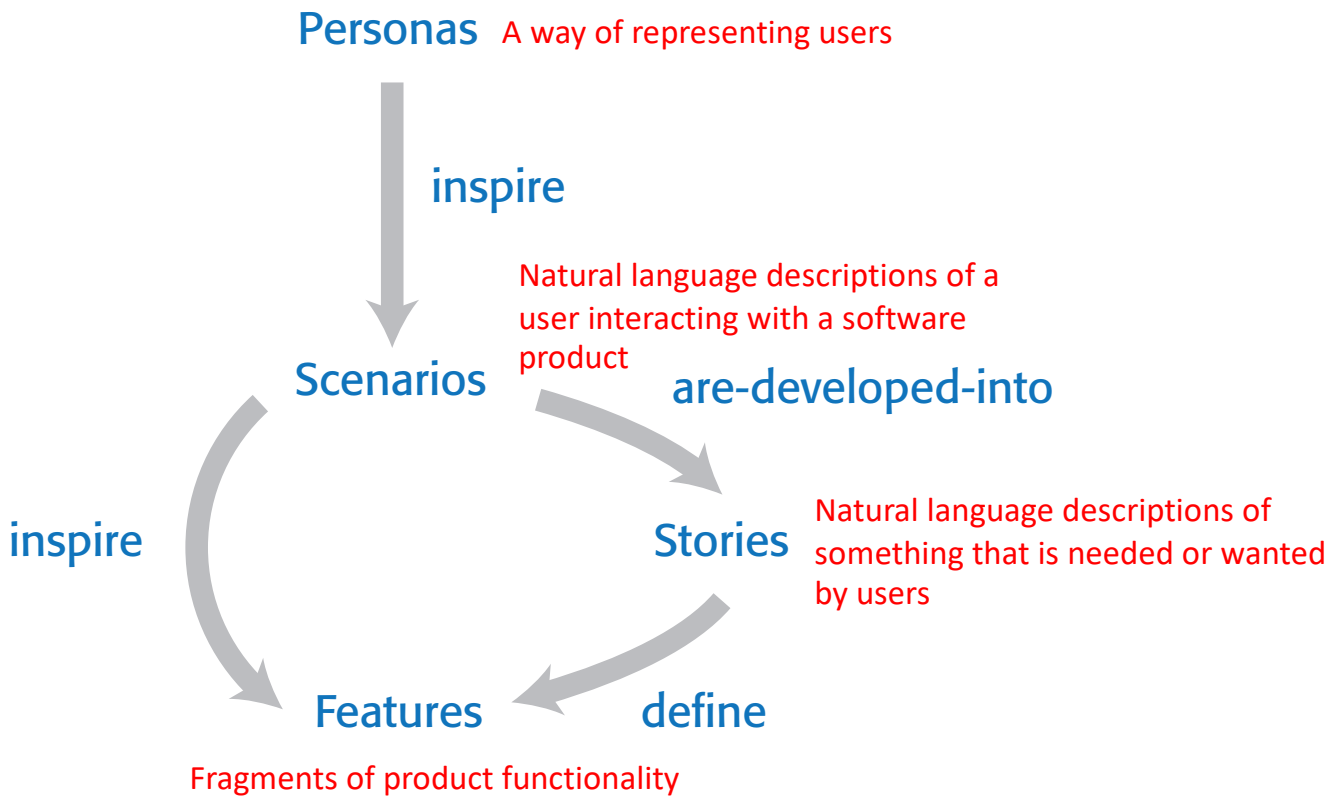



Personas

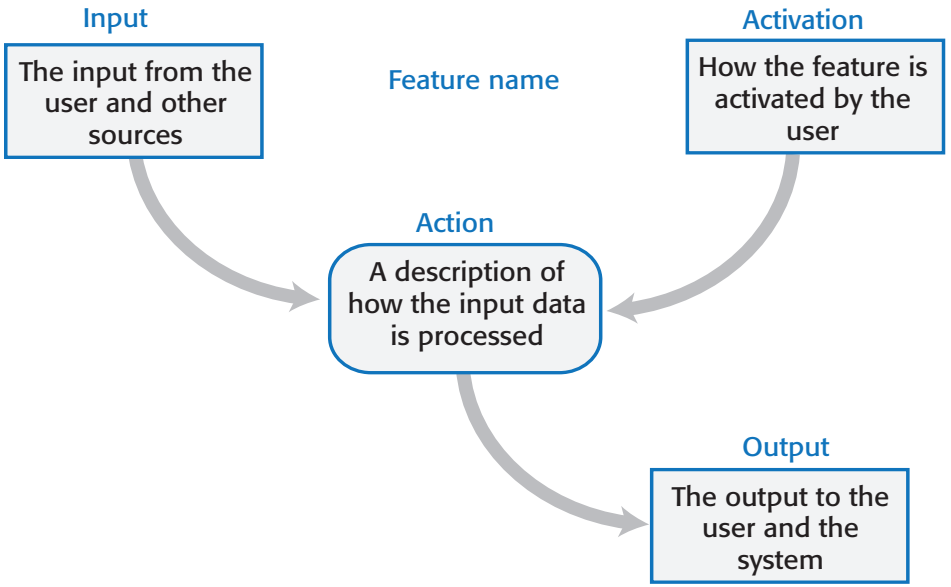
User Understanding

- It makes sense in any product development to spend time trying to understand the potential users and customers of your product.
- A range of techniques have been developed for understanding the ways that people work and use software.
  - These include user interviews, surveys, ethnography and task analysis.
  - Some of these techniques are expensive and unrealistic for small companies.
- Informal user analysis and discussions, which simply involve asking users about their work, the software that they use, and its strengths and weaknesses are inexpensive and very valuable.

From Personas to Features



Feature Description



Personas

- You need to have an understanding of your potential users to design features that they are likely to find useful and to design a user interface that is suited to them.
- Personas are 'imagined users' where you create a character portrait of a type of user that you think might use your product.
  - For example, if your product is aimed at managing appointments for dentists, you might create a dentist persona, a receptionist persona and a patient persona.
- Personas of different types of user help you imagine what these users may want to do with your software and how it might be used. They help you envisage difficulties that they might have in understanding and using product features.

A Persona for a Primary School Teacher

Jack, a primary school teacher

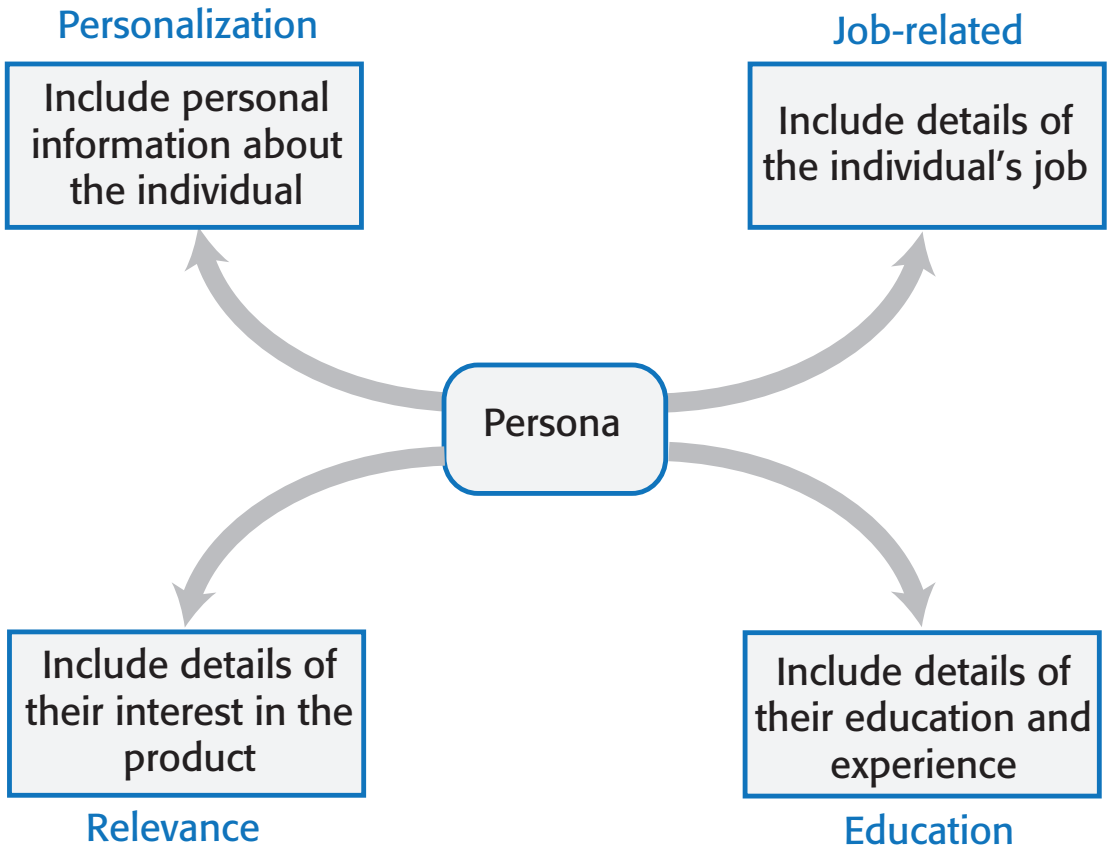
Jack, age 32, is a primary school (elementary school) teacher in Ullapool, a large coastal village in the Scottish Highlands. He teaches children from ages 9-12. He was born in a fishing community north of Ullapool, where his father runs a marine fuels supply business and his mother is a community nurse. He has a degree in English from Glasgow University and retrained as a teacher after several years working as a web content author for a large leisure group.

Jack's experience as a web developer means that he is confident in all aspects of digital technology. He passionately believes that the effective use of digital technologies, blended with face to face teaching, can enhance the learning experience for children. He is particularly interested in using the iLearn system for project-based teaching, where students work together across subject areas on a challenging topic.

Persona Descriptions

- A persona should ‘paint a picture’ of a type of product user. They should be relatively short and easy-to-read.
- You should describe their background and why they might want to use your product.
- You should also say something about their educational background and technical skills.
- These help you assess whether or not a software feature is likely to be useful, understandable and usable by typical product users.

Persona Descriptions



Aspects of a persona description

Personalization
You should give them a name and say something about their personal circumstances. This is important because you shouldn't think of a persona as a role but as an individual. It is sometimes helpful to use an appropriate stock photograph to represent the person in the persona. Some studies suggest that this helps project teams use personas more effectively.

Job-related
If your product is targeted at business, you should say something about their job and (if necessary) what that job involves. For some jobs, such as a teacher where readers are likely to be familiar with the job, this may not be necessary.

Education
You should describe their educational background and their level of technical skills and experience. This is important, especially for interface design.

Relevance
If you can, you should say why they might be interested in using the product and what they might want to do with it.

Emma, a history teacher

Emma, age 41, is a history teacher in a secondary school (high school) in Edinburgh. She teaches students from ages 12 to 18. She was born in Cardiff in Wales where both her father and her mother were teachers. After completing a degree in history from Newcastle University, she moved to Edinburgh to be with her partner and trained as a teacher. She has two children, aged 6 and 8, who both attend the local primary school. She likes to get home as early as she can to see her children, so often does lesson preparation, administration and marking from home.

Emma uses social media and the usual productivity applications to prepare her lessons, but is not particularly interested in digital technologies. She hates the virtual learning environment that is currently used in her school and avoids using it if she can. She believes that face-to-face teaching is most effective. She might use the iLearn system for administration and access to historic films and documents. However, she is not interested in a blended digital/face-to-face approach to teaching.

Elena, a school IT technician

Elena, age 28, is a senior IT technician in a large secondary school (high school) in Glasgow with over 2000 students. Originally from Poland, she has a diploma in electronics from Potsdam University. She moved to Scotland in 2011 after being unemployed for a year after graduation. She has a Scottish partner, no children, and hopes to develop her career in Scotland. She was originally appointed as a junior technician but was promoted, in 2014, to a senior post responsible for all the school computers.

Although not involved directly in teaching, Elena is often called on to help in computer science classes. She is a competent Python programmer and is a ‘power user’ of digital technologies. She has a long-term career goal of becoming a technical expert in digital learning technologies and being involved in their development. She wants to become an expert in the iLearn system and sees it as an experimental platform for supporting new uses for digital learning.

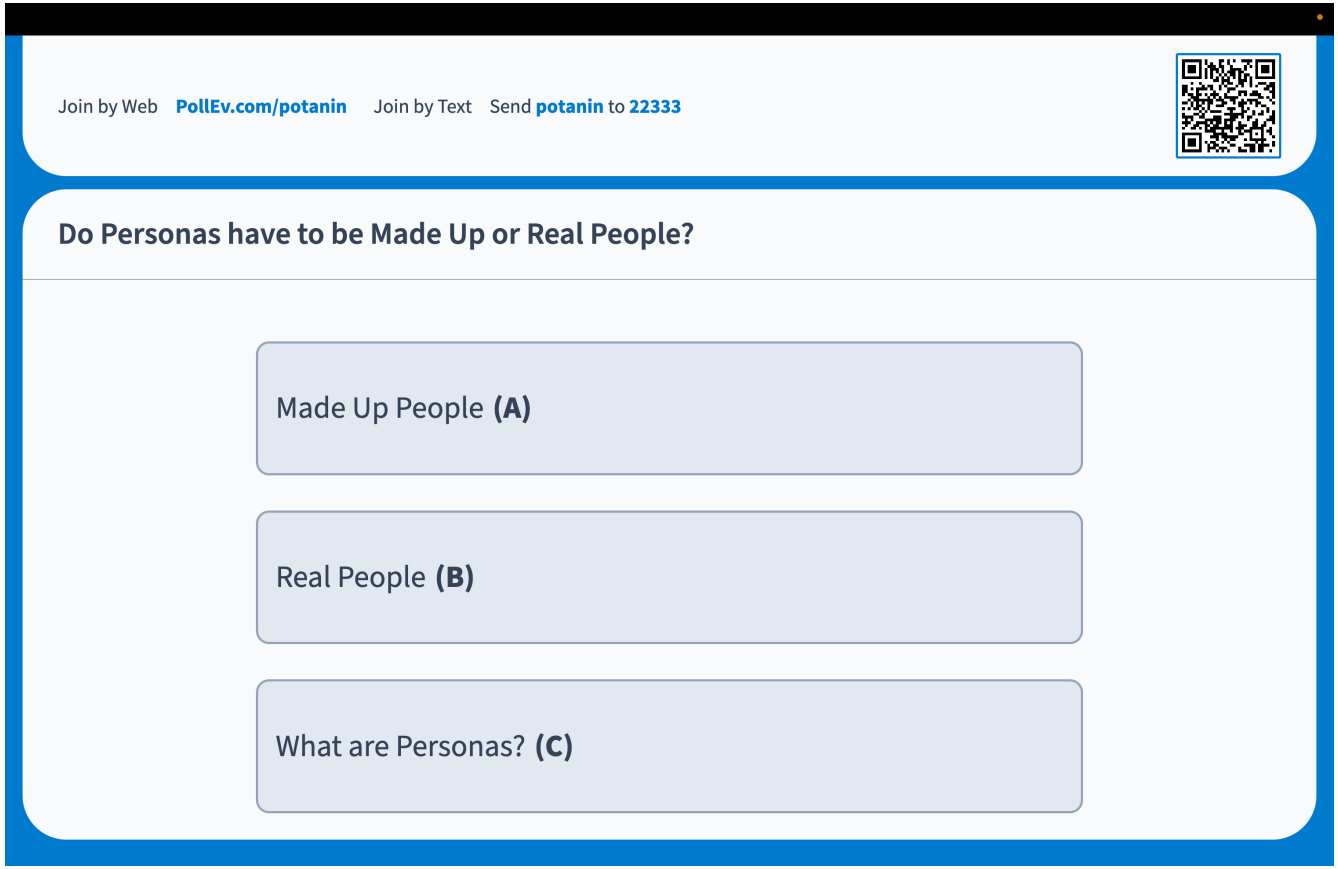
Persona Benefits

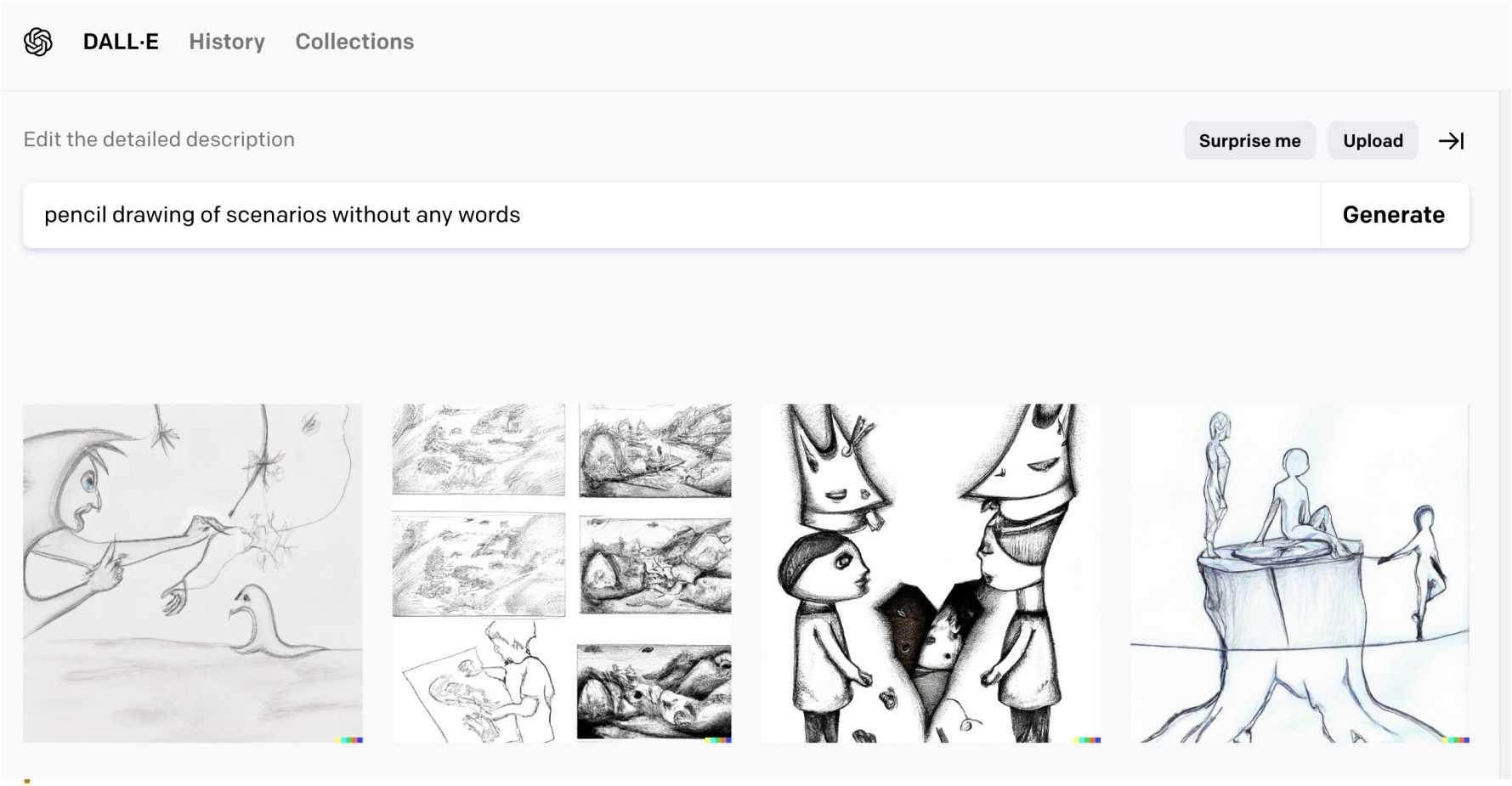
- The main benefit of personas is that they help you and other development team members empathize with potential users of the software.
- Personas help because they are a tool that allows developers to 'step into the user's shoes'.
  - Instead of thinking about what you would do in a particular situation, you can imagine how a persona would behave and react.
- Personas can help you check your ideas to make sure that you are not including product features that aren't really needed.
- They help you to avoid making unwarranted assumptions, based on your own knowledge, and designing an over-complicated or irrelevant product.

Deriving Personas

- Personas should be based on an understanding of the potential product users, their jobs, their background and their aspirations.
- You should study and survey potential users to understand what they want and how they might use the product.
- From this data, you can then abstract the essential information about the different types of product user and use this as a basis for creating personas.
- Personas that are developed on the basis of limited user information are called proto-personas.
  - Proto-personas may be created as a collective team exercise using whatever information is available about potential product users. They can never be as accurate as personas developed from detailed user studies, but they are better than nothing.

Poll Everywhere Time!





Scenarios

Scenarios

- A scenario is a narrative that describes how a user, or a group of users, might use your system.
- There is no need to include everything in a scenario – the scenario isn't a system specification.
- It is simply a description of a situation where a user is using your product's features to do something that they want to do.
- Scenario descriptions may vary in length from two to three paragraphs up to a page of text.

Jack's scenario: using the iLearn system for class projects

Fishing in Ullapool

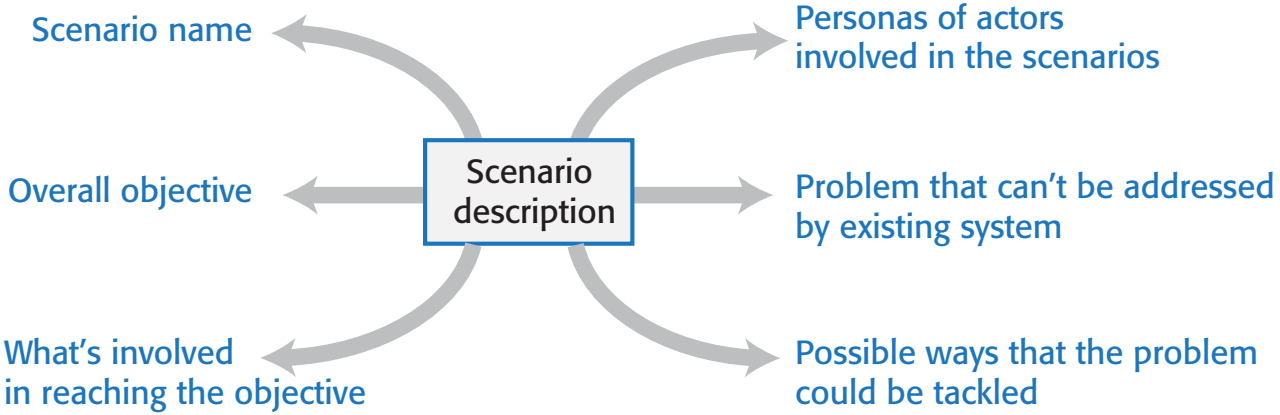
Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history archive site) to access newspaper archives and photographs. However, Jack also needs a photo-sharing site as he wants students to take and comment on each others' photos and to upload scans of old photographs that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack sends an email to a primary school teachers' group to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account with KidsTakePics.

He uses the the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that, when they log in, they can immediately use the system to upload photos from their phones and class computers.

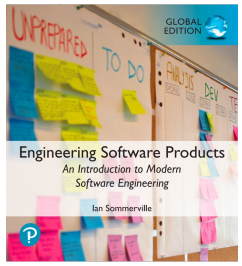
Elements of a scenario description



Scenario elements

- A brief statement of the overall objective.
  - In Jack's scenario, this is to support a class project on the fishing industry.
- References to the personas involved (Jack) so that you can get information about the capabilities and motivation of that user.
- Information about what is involved in doing the activity. For example, in Jack's scenario this involves gathering reminiscences from relatives, accessing newspaper archives, etc.
- An explanation of problems that can't be readily addressed using the existing system.
  - Young children don't understand issues such as copyright and privacy, so photo sharing requires a site that a teacher can moderate to make sure that published images are legal and acceptable.
- A description of one way that the identified problem might be addressed.
  - In Jack's scenario, the preferred approach is to use an external tool designed for school students.

Emma's scenario



- Emma's scenario is different from Jack's scenario in that it describes a common and well-understood process rather than something new.
- Emma is an e-learning sceptic and she is not interested in innovative applications. She wants a system that will make her life easier and reduce the amount of routine administration that she has to do.
- The scenario discusses how parts of the process (setting up an email group and web page) are automated by the iLearn system.

Emma's scenario: using iLearn for administration

Emma is teaching the history of the First World War to a class of 14 year olds (S3). A group of S3 students are visiting the historic World War One battlefields in northern France. She want to set up a 'battlefields group' where the students who are attending the trip can share their research about the places they are visiting as well as their pictures and thoughts about the visit.

From home, she logs onto the iLearn system system using her Google account credentials. Emma has two iLearn accounts – her teacher account and a parent account associated with the local primary school. The system recognises that she is a multiple account owner and asks her to select the account to be used. She chooses the teacher account and the system generates her personal welcome screen. As well as her selected applications, this also shows management apps that help teachers create and manage student groups.

Emma selects the 'group management' app, which recognizes her role and school from her identity information and creates a new group. The system prompts for the class year (S3) and subject (history) and automatically populates the new group with all S3 students who are studying history. She selects those students going on the trip and adds her teacher colleagues, Jamie and Claire, to the group.

Emma's scenario: using iLearn for administration

She names the group and confirms that it should be created. The app sets up an icon on her iLearn screen to represent the group, creates an email alias for the group and asks Emma if she wishes to share the group. She shares access with everyone in the group, which means that they also see the icon on their screen. To avoid getting too many emails from students, restricts sharing of the email alias to Jamie and Claire.

The group management app then asksEmma if she wishes to set up a group web page, wiki and blog. Emma confirms that a web page should be created and she types some text to be included on that page.

She then accesses flickr using the icon on her screen, logs in and creates a private group to share trip photos that students and teachers have taken. She uploads some of her own photos from previous trips and emails an invitation to join the photo-sharing group to the Battlefield email list. Emma uploads material from her own laptop that she has written about the trip to iLearn and shares this with the 'Battlefields Group'. This action adds her documents to the web page and generates an alert to group members that new material is available.

Writing scenarios

- Scenarios should always be written from the user's perspective and based on identified personas or real users.
- Your starting point for scenario writing should be the personas that you have created. You should normally try to imagine several scenarios from each persona.
- Ideally, scenarios should be general and should not include implementation information.
  - However, describing an implementation is often the easiest way to explain how a task is done.
- It is important to ensure that you have coverage of all of the potential user roles when describing a system.

Elena's scenario: configuring the iLearn system

Elena has been asked by David, the head of the art department in her school, to help set up an iLearn environment for his department. David wants an environment that includes tools for making and sharing art, access to external websites to study artworks, and 'exhibition' facilities so that the students' work can be displayed.

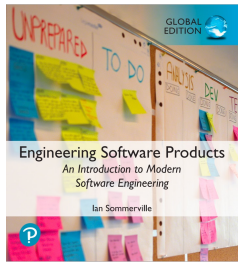
Elena's starts by talking to art teachers to discover the tools that they recommend and the art sites that they use for studies. She also discovers that the tools they use and the sites they access vary according to the age of their students. Consequently, different student groups should be presented with a toolset that is appropriate for their age and experience.

Once she has established what is required, Elena logs into the iLearn system as an administrator and starts configuring the art environment using the iLearn setup service. She creates sub-environments for three age groups plus a shared environment that includes tools and sites that may be used by all students.

She drags and drops tools that are available locally and the URLs of external websites into each of these environments. For each of the sub-environments, she assigns an art teacher as its administrator so that they can refine the tool and web site selection that has been set up. She publishes the environments in 'review mode' and makes them available to the teachers in the art department.

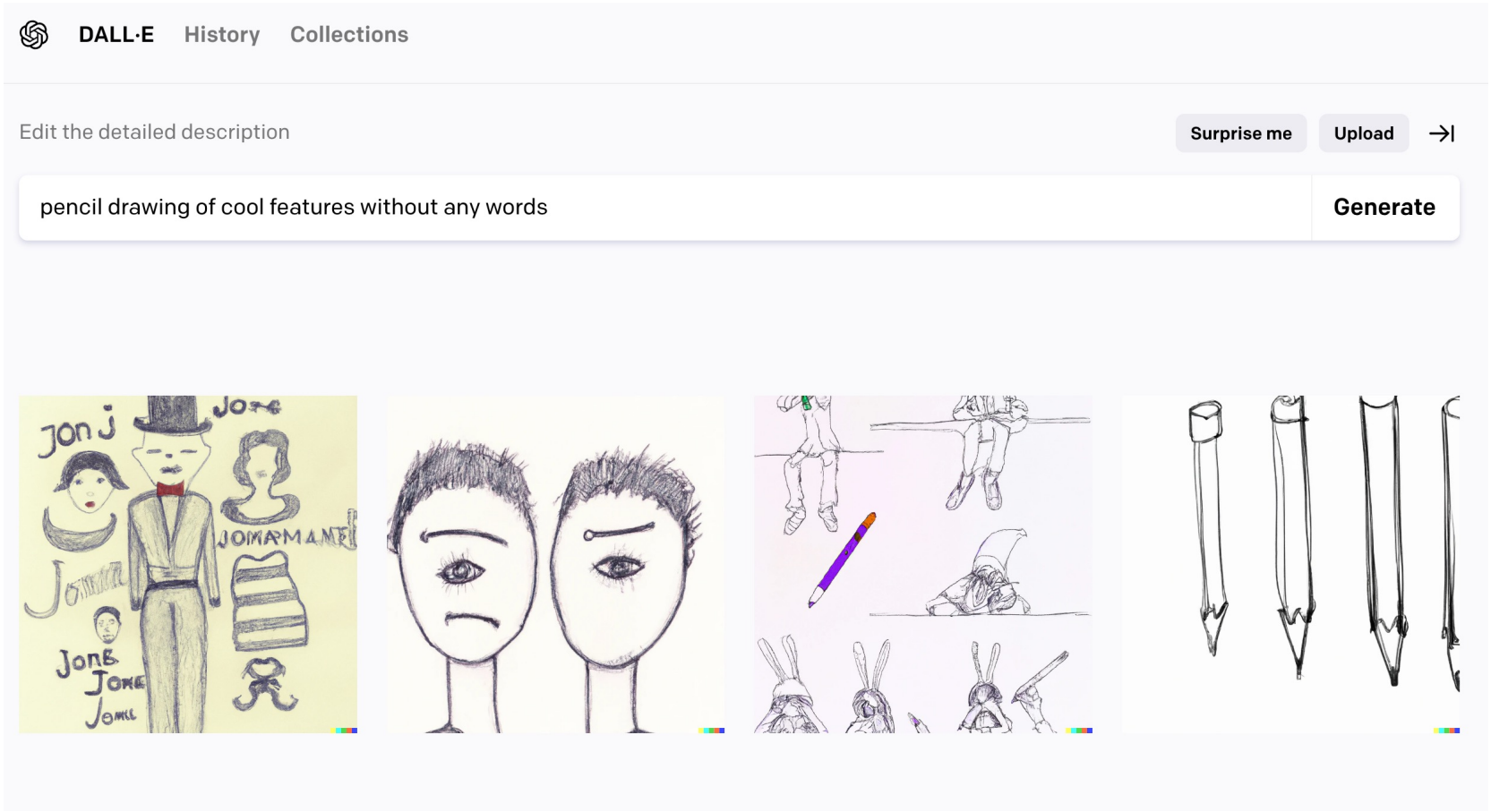
After discussing the environments with the teachers, Elena shows them how to refine and extend the environments. Once they have agreed that the art environment is useful, it is released to all students in the school.

User involvement



- It is easy for anyone to read and understand scenarios, so it is possible to get users involved in their development.
- The best approach is to develop an imaginary scenario based on our understanding of how the system might be used then ask users to explain what you have got wrong.
- They might ask about things they did not understand and suggest how the scenario could be extended and made more realistic.
- Our experience was that users are not good at writing scenarios.
  - The scenarios that they created were based on how they worked at the moment. They were far too detailed and the users couldn't easily generalize their experience.

Poll Everywhere Time!



Features

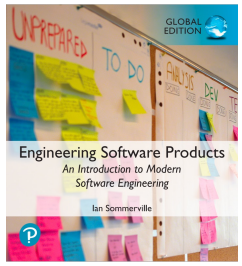
Software Features

- A feature is a fragment of functionality such as a ‘print’ feature, a ‘change background feature’, a ‘new document’ feature and so on.
- Before you start programming a product, you should aim to create a list of features to be included in your product.
- The feature list should be your starting point for product design and development.
- But remember that there are overall *goals*, as well as *functional requirements* versus *quality requirements!*

User stories

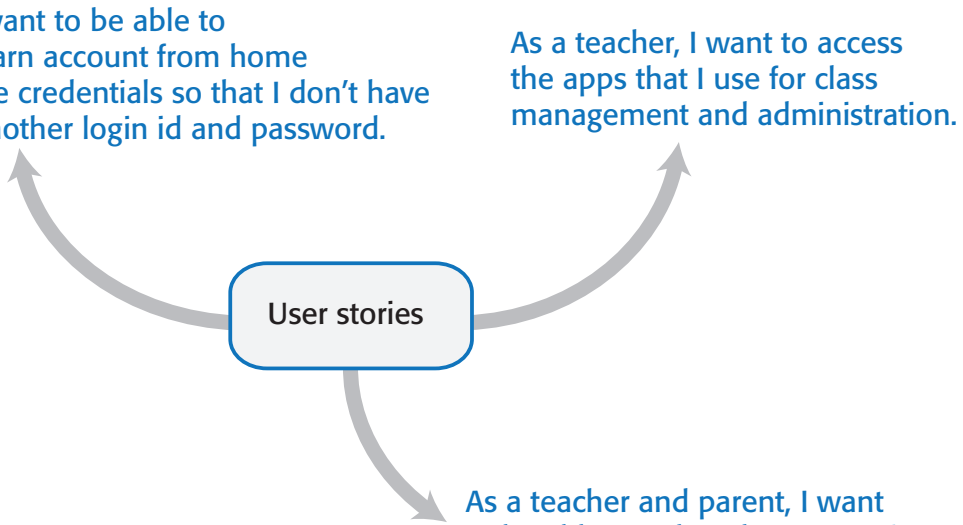
- Scenarios are high-level stories of system use. They should describe a sequence of interactions with the system but should not include details of these interactions.
- User stories are finer-grain narratives that set out in a more detailed and structured way a single thing that a user wants from a software system.
  - As an author, I need a way to organize the book that I'm writing into chapters and sections.
- This story reflects what has become the standard format of a user story:
  - **As a** <role>, I <want | need> **to** <do something>
    - As a teacher, I want to tell all members of my group when new information is available
- A variant of this standard format adds a justification for the action:
  - **As a** <role> I <want | need> **to** <do something> **so that** <reason>
    - As a teacher, I need to be able to report who is attending a class trip so that the school maintains the required health and safety records.

User stories in planning

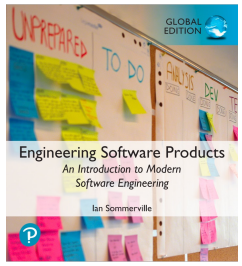


- An important use of user stories is in planning.
  - Many users of the Scrum method represent the product backlog as a set of user stories.
- User stories should focus on a clearly defined system feature or aspect of a feature that can be implemented within a single sprint.
- If the story is about a more complex feature that might take several sprints to implement, then it is called an epic.
  - As a system manager, I need a way to backup the system and restore either individual applications, files, directories or the whole system.
  - There is a lot of functionality associated with this user story. For implementation, it should be broken down into simpler stories with each story focusing on a single aspect of the backup system.

User stories from Emma's scenario

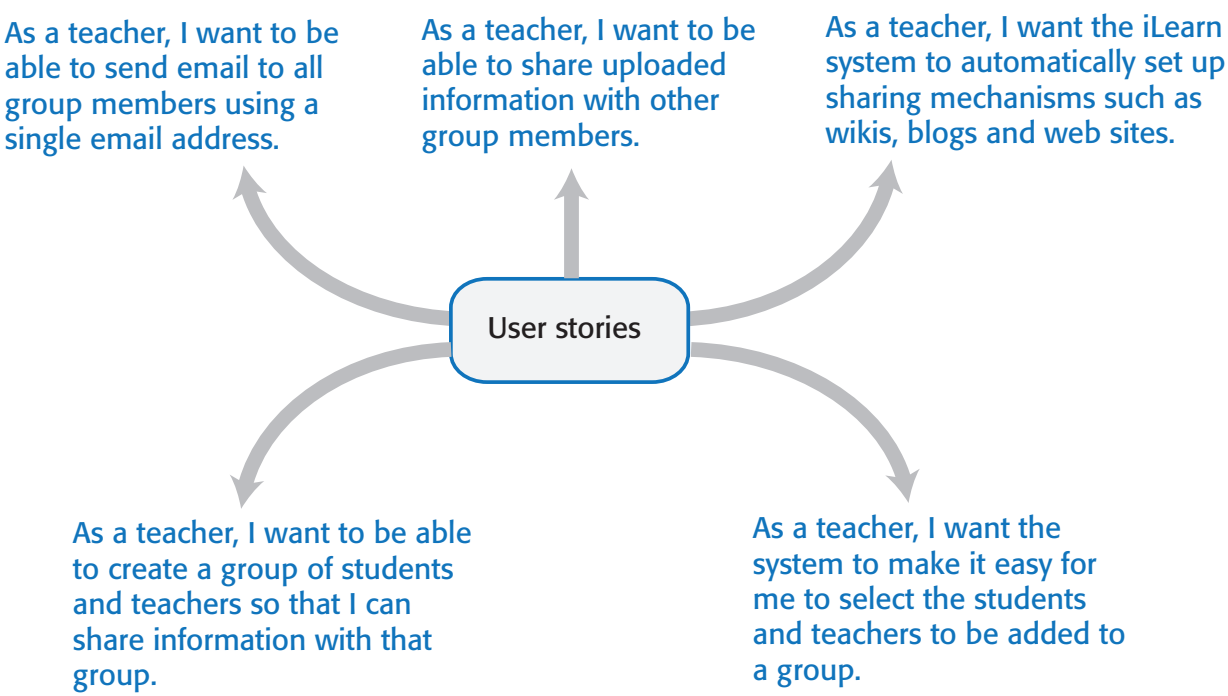


Feature description using user stories



- Stories can be used to describe features in your product that should be implemented.
- Each feature can have a set of associated stories that describe how that feature is used.

User stories describing the Groups feature



Stories and scenarios

- As you can express all of the functionality described in a scenario as user stories, do you really need scenarios?'

- Scenarios are more natural and are helpful for the following reasons:

  - Scenarios read more naturally because they describe what a user of a system is actually doing with that system. People often find it easier to relate to this specific information rather than the statement of wants or needs set out in a set of user stories.

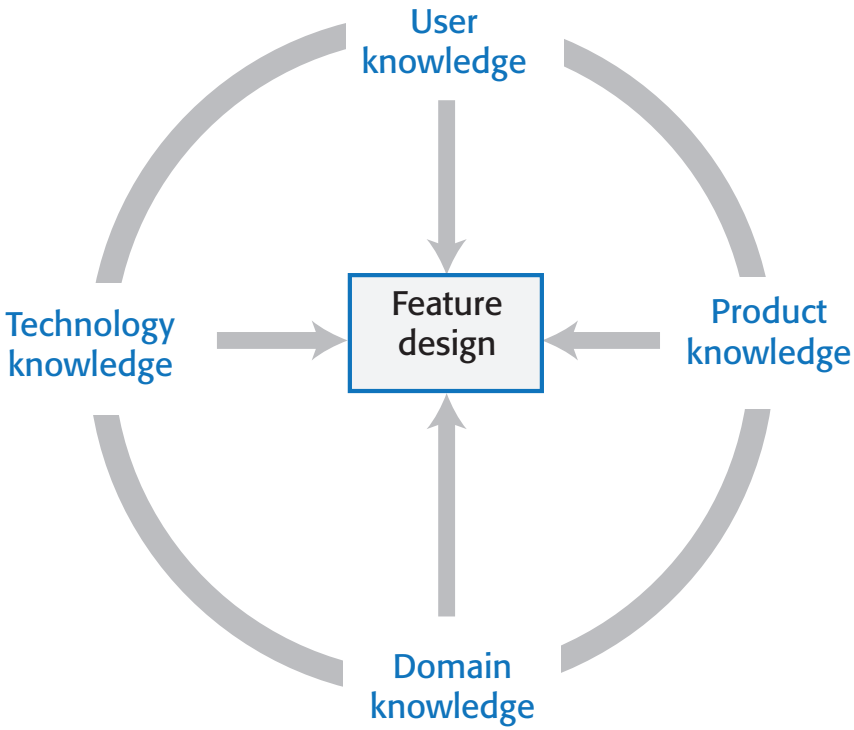
  - If you are interviewing real users or are checking a scenario with real users, they don't talk in the stylized way that is used in user stories. People relate better to the more natural narrative in scenarios.

  - Scenarios often provide more context - information about what the user is trying to do and their normal ways of working. You can do this in user stories, but it means that they are no longer simple statements about the use of a system feature.

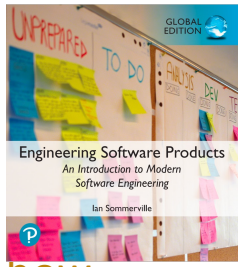
Feature identification

- Your aim in the initial stage of product design should be to create a list of features that define your product.
- A feature is a way of allowing users to access and use your product's functionality so the feature list defines the overall functionality of the system.
- Features should be independent, coherent and relevant:
  - *Independence*
    Features should not depend on how other system features are implemented and should not be affected by the order of activation of other features.
  - *Coherence*
    Features should be linked to a single item of functionality. They should not do more than one thing and they should never have side-effects.
  - *Relevance*
    Features should reflect the way that users normally carry out some task. They should not provide obscure functionality that is hardly ever required.

Feature design

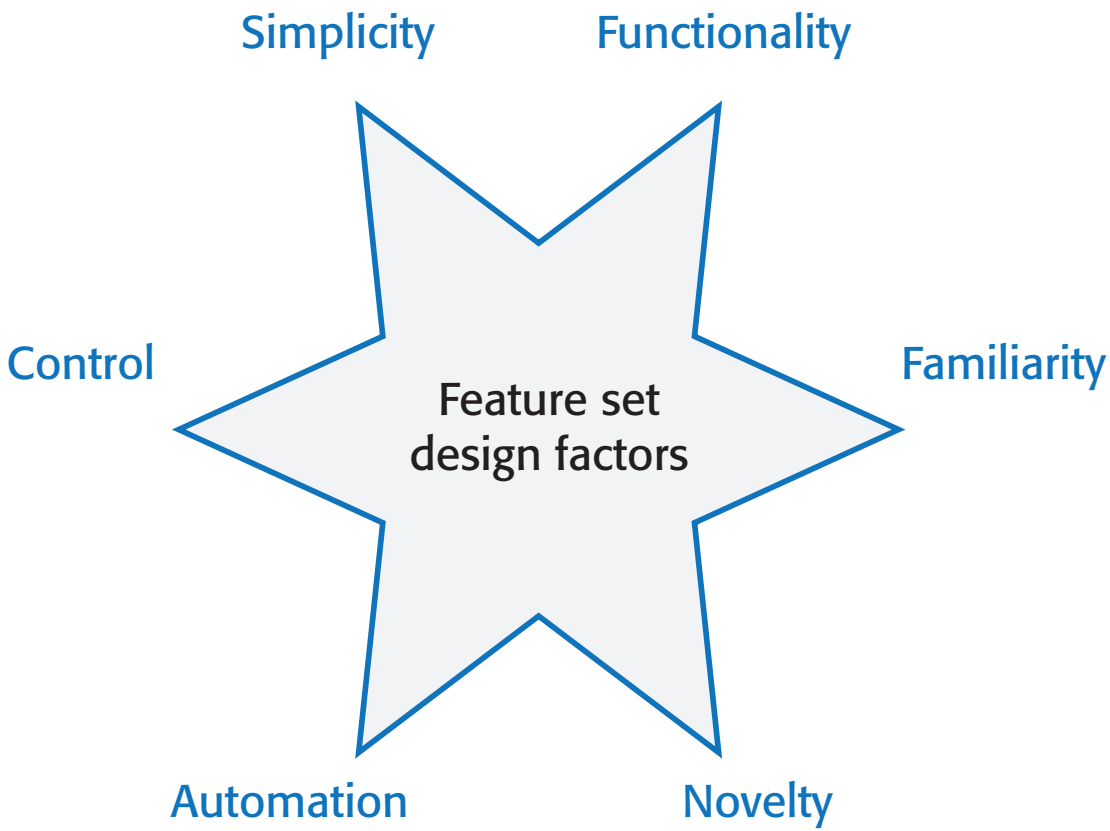


Knowledge required for feature design



- ***User knowledge***
  You can use user scenarios and user stories to inform the team of what users want and how they might use it the software features.
- ***Product knowledge***
  You may have experience of existing products or decide to research what these products do as part of your development process. Sometimes, your features have to replicate existing features in these products because they provide fundamental functionality that is always required.
- ***Domain knowledge***
  This is knowledge of the domain or work area(e.g. finance, event booking) that your product aims to support. By understanding the domain, you can think of new innovative ways of helping users do what they want to do.
- ***Technology knowledge***
  New products often emerge to take advantage of technological developments since their competitors were launched. If you understand the latest technology, you can design features to make use of it.

Factors in feature set design



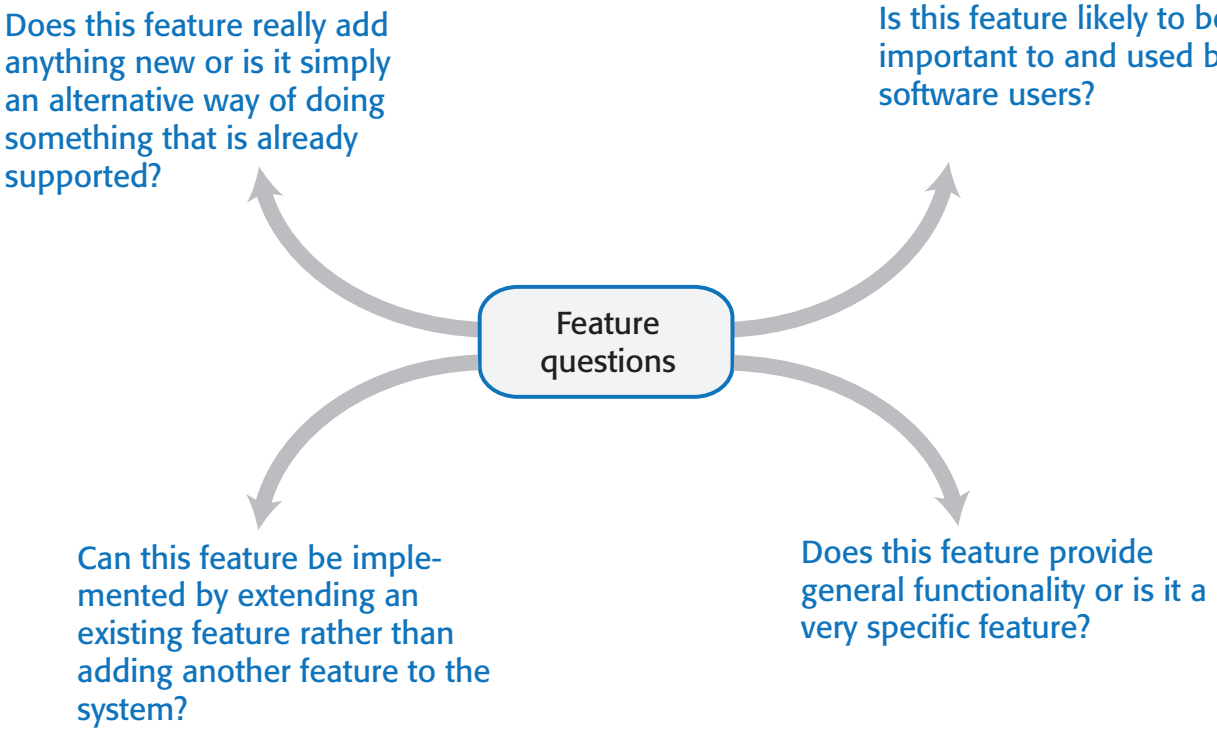
Feature trade-offs

- Simplicity and functionality
  - You need to find a balance between providing a simple, easy-to-use system and including enough functionality to attract users with a variety of needs.
- Familiarity and novelty
  - Users prefer that new software should support the familiar everyday tasks that are part of their work or life. To encourage them to adopt your system, you need to find a balance between familiar features and new features that convince users that your product can do more than its competitors.
- Automation and control
  - Some users like automation, where the software does things for them. Others prefer to have control. You have to think carefully about what can be automated, how it is automated and how users can configure the automation so that the system can be tailored to their preferences.

Feature creep

- Feature creep occurs when new features are added in response to user requests without considering whether or not these features are generally useful or whether they can be implemented in some other way.
- Too many features make products hard to use and understand
- There are three reasons why feature creep occurs:
  - Product managers are reluctant to say ‘no’ when users ask for specific features.
  - Developers try to match features in competing products.
  - The product includes features to support both inexperienced and experienced users.

Factors in feature set design



Feature derivation

- Features can be identified directly from the product vision or from scenarios.
- You can highlight phrases in narrative description to identify features to be included in the software.
  - You should think about the features needed to support user actions, identified by active verbs, such as use and choose.

The iLearn system vision

- FOR teachers and educators WHO need a way *to help students use web-based learning resources and applications*, THE iLearn system is an open learning environment THAT *allows the set of resources used by classes and students to be easily configured for these students and classes by teachers themselves*.
- UNLIKE Virtual Learning Environments, such as Moodle, the focus of iLearn is the learning process itself, rather than the administration and management of materials, assessments and coursework. OUR product *enables teachers to create subject and age-specific environments for their students* using any web-based resources, such as videos, simulations and written materials that are appropriate

Features from the product vision

- A feature that allows users to access and use existing web-based resources;
- A feature that allows the system to exist in multiple different instantiations;
- A feature that allows user configuration of the system to create a specific instantiation.

Jack's scenario with highlighted phrases

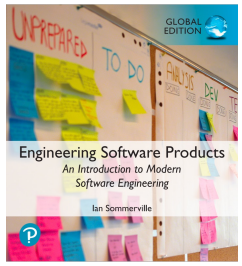
Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. *Students use an iLearn wiki* to gather together fishing stories and *SCRAN (a history archive) to access newspaper archives and photographs.* However, Jack also needs a photo-sharing site as he wants *pupils to take and comment on each others' photos* and to *upload scans of old photographs* that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack *sends an email to a primary school teachers' group*, which he is a member of to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics *is not integrated with the iLearn authentication service*, he sets up a teacher and a class account with KidsTakePics.

He uses the the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that when they log in, they can immediately use the system to upload photos from their phones and class computers.

Features from Jack's scenario



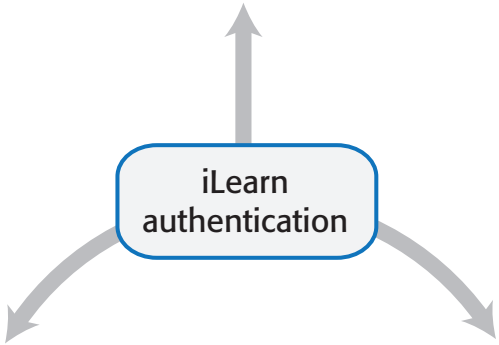
- A wiki for group writing.
- Access to the SCRAN history archive. This is a shared national resource that provides access to historical newspaper and magazine articles for schools and universities.
- Features to set up and access an email group.
- A feature to integrate applications with the iLearn authentication service.

The feature list

- The output of the feature identification process should be a list of features that you use for designing and implementing your product.
- There is no need to go into a lot of detail about the features at this stage. You add detail when you are implementing the feature.
- You can describe features using a standard input-action-output template by using structured narrative descriptions or by a set of user stories.

The iLearn authentication feature

Description
Authentication is used to identify users to the system and is currently based on a login id/password system. Users may authenticate themselves using their national user id and a personal password or may use their Google or Facebook credentials.



Constraints
All users must have a national user id and system password that they use for initial system authentication. They may then link their account with their Google/Facebook account for future authentication sessions.

Comments
Future authentication mechanisms may be based on biometrics and this should be considered in the design of the system.

Feature description using user stories

- ***Description***
  As a system manager, I want to create and configure an iLearn environment by adding and removing services to/from that environment so that I can create environments for specific purposes.

- As a system manager, I want to set up sub-environments that include a subset of services that are included in another environment.

- As a system manager, I want to assign administrators to created environments.

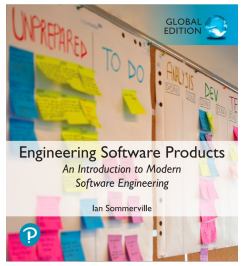
- As a system manager, I want to limit the rights of environment administrators so that they cannot accidentally or deliberately disrupt the operation of key services.

- As a teacher, I want to be able to add services that are not integrated with the iLearn authentication system.

- ***Constraints***
  The use of some tools may be limited for license reasons so there may be a need to access license management tools during configuration.

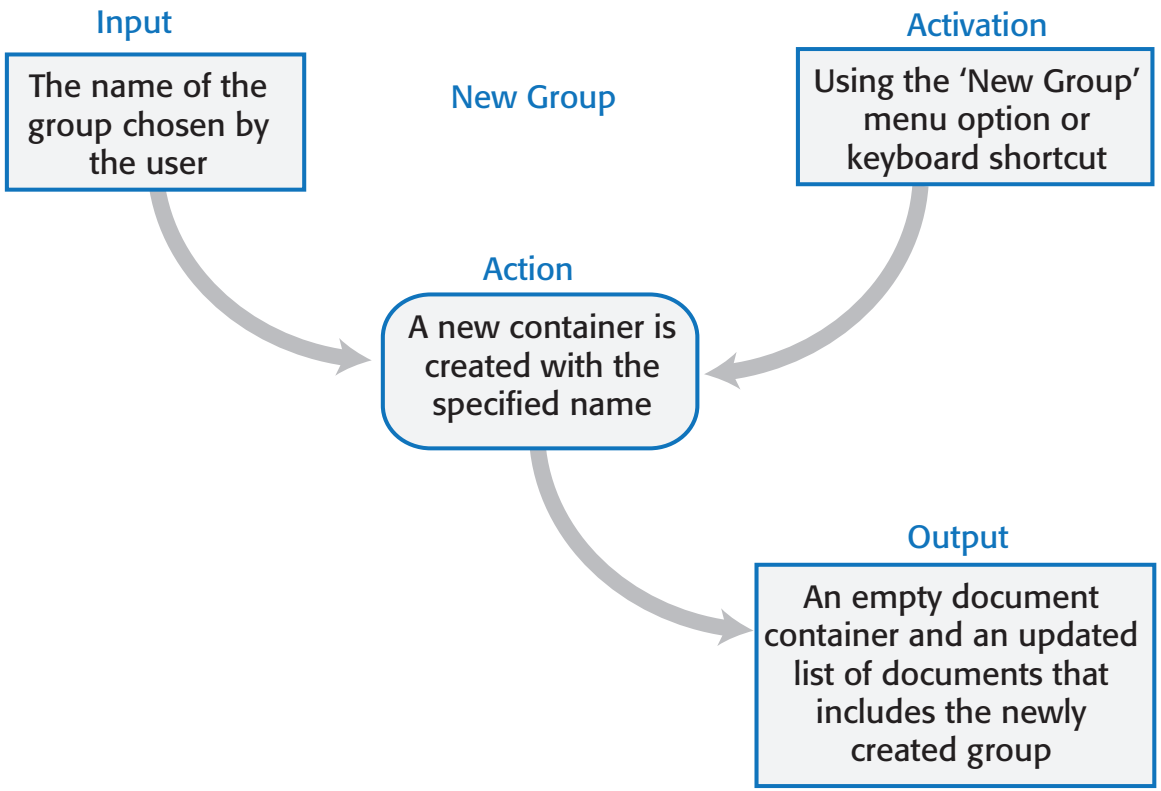
- ***Comments***
  Based on Elena's and Jack's scenarios

Innovation and feature identification



- Scenarios and user stories should always be your starting point for identifying product features.
  - Scenarios tell you how users work at the moment. They don't show how they might change their way of working if they had the right software to support them.
  - Stories and scenarios are 'tools for thinking' and they help you gain an understanding of how your software might be used. You can identify a feature set from stories and scenarios.
- User research, on its own, rarely helps you innovate and invent new ways of working.
- You should also think creatively about alternative or additional features that help users to work more efficiently or to do things differently.

The “New Group” Feature Description



Poll Everywhere Time!