Expanded Guidance – English version

User stories

User stories were originally developed as means of describing requirements of a product from the perspective of the user of that product – hence the name "User Story". The technique was popularised by Mike Cohn in his book "*User Stories Applied*" and is now one of the most popular ways of expressing Product Backlog items world-wide.

Format

User Stories have a specific format:

> **As a** <user/beneficiary>
> **I want** <this feature / capability>
> **So that** <I can achieve this goal>

This format is considered valuable because it builds on the simple statement of what is needed (the "*I want*…") by providing context and rationale for the need in the "*As a*…" and "*So that*…" statements to help shape understanding of the need and to help shape a solution to meet that need.

The Scrum Guide states that "*A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract"* and that *"The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against"*

The format of a User Story clearly provides a description of a goal or a need within a boundary (in the I want… statement), well defined users or customers (As a… statement), and a focus on the delivery of value (in the So that… statement). Collectively the statements provide the Scrum Team with both the need and justification for a future state of a product to plan against and, in the descriptions below, show how the essential journey from Product Backlog item to valuable Increment is facilitated by properly structured User Stories.

User Stories and the Product Backlog

Product Backlog Items can be expressed as stories at all levels from the top-level product goal – for example:

> As a *travelling salesperson*
>
> I want *a motor vehicle for my personal use*
>
> So that *I can transport myself and a variety of illustrations and samples from one potential customer to another in a given day*

Through more specific goals associated with subsets of the overarching product for example:

> As a *travelling salesperson*
>
> I want *a satellite navigation system in my motor vehicle*
>
> So that *I can plan a route that allows me to maximise the number of customers that I can visit in a single day*

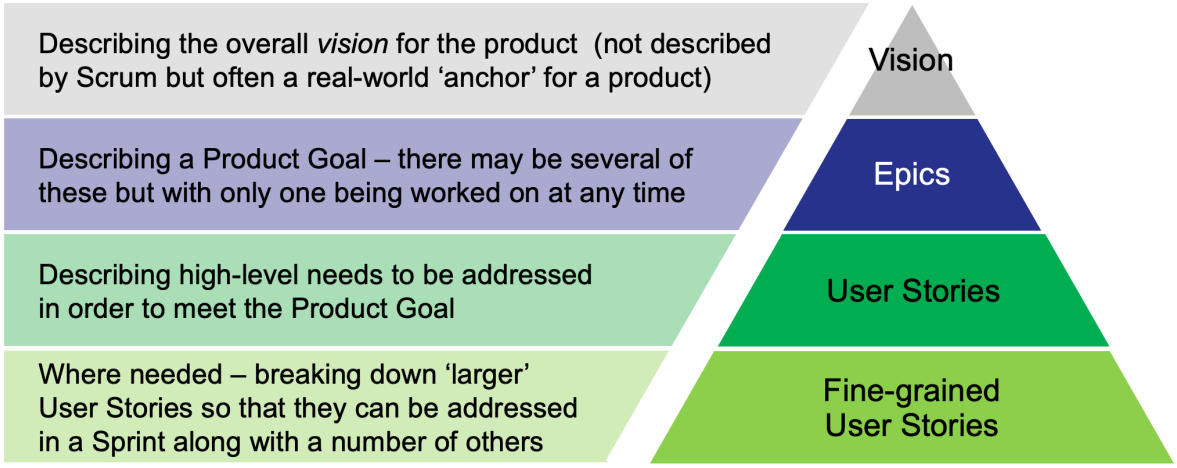
Down to a specific detailed goal of that for example.

> As a *travelling salesperson*
>
> I want *live travel information available to my vehicle's satellite navigation system*

> So that *I can plan the most time-efficient route from one customer site to the next and maximise the number of customers that I can visit in a single day*

In a Scrum context, stories in the Product Backlog get broken down into increasingly fine-grained stories, in the process of *backlog refinement*, until they are small enough to be addressed within a single Sprint.

Stories at different levels are often given different labels such as those shown in the diagram below. In Scrum terms, whatever their label these are simply Product Goals and Product Backlog Items expressed at different levels of granularity within the Product Backlog.



Composition of a User Story

A Backlog item that is expressed clearly and concisely and is small enough to become done within the boundaries of a single Sprint is considered ***ready*** for acceptance into that Sprint.

In addition, Scrum requires at least one Increment to be delivered in each Sprint with each Increment meeting its Definition of Done commitment. Given that an Increment is born the moment that a Product Backlog item meets its Definition of Done, it is clear that such a definition needs to be associated with each backlog item as part of it being *ready.* Expressing Product Backlog items as User Stories aligns very well with the processes associated with creating, refining and ordering the backlog but to work optimally requires the Scrum Team to go beyond the simplicity of the 'As a… I want… So that' structure.

When the User Story technique was created, stories were considered to be made up of three elements, known as the Three C's.

- The **Card** on which the story is written
- The **Conversation** required to achieve a collective understanding of the story
- The **Confirmation** that provides a Definition of Done for the story

More recently some practitioners have found it valuable to consider a fourth C:

- The **Context** within which the story, or the Increment that it will become, exists

The Card

User Stories are intended to be captured on an index card of dimensions 6 inches by 4 inches (approximately 15cm by 10cm). Restricting the space available intentionally constrains the detail

of the need that gets captured, ensuring that it is expressed concisely. In addition, the physical card acts a 'token for a conversation' and is often used as a token in planning sessions to help create a sensible order for the Product Backlog.

The Conversation

The Product Backlog is an *emergent, ordered* list of what is needed to improve the product. Product Backlog refinement is the activity that manages the emergence of detail, the order that Product Backlog items should be addressed in and ensures that those items are ready to be accepted into an upcoming Sprint. It is a recurring, collaborative effort involving the entire Scrum Team with both ordering and understanding of individual items coming from the *conversation* of all involved. It makes sense to defer discussing details of Product Backlog items to the last sensible point before their inclusion in a Sprint in order to ensure that decisions about the emergent product are based on the most up-to-date information available

The Confirmation

One of the characteristics of a Product Backlog item that is *ready* for inclusion in a Sprint is that its Definition of Done is clear. The Confirmation element of User Story is its Definition of Done. It poses questions by which the Product Owner can 'confirm' that the story has been 'Done' to the extent that the user/beneficiary (As a…) could use the feature/capability of the Increment (I want…) to achieve their goal (So that…)

The Context

Providing a Context for a story essentially describes how it relates to other stories and may prove useful both in terms of refining understanding of an individual Product Backlog item and the order in which items should be addressed. It may also help establish an understanding of how a number of related stories can help form and/or fulfil a Sprint Goal. Various *modelling* techniques may be employed in the process of identifying and refining stories required to achieve Product Goals and also to help ensure coherence of the Product itself as Increments are delivered "as concrete stepping stones towards the Product Goal". In regulated environments such models may be formally constituted and controlled, in less formal environments they may simply reflect the clustering of story cards on a table or on a wall. An often-used example of this is the technique of Story Mapping by Jeff Patton.

Connecting Lean thinking with User Stories, Backlogs, Sprints and Increments

Lean thinking encourages us *to reduce waste and focus on the essentials* of the work we are doing and to continuously improve our processes to achieve this. It encourage us to focus on:

Value – for example by specifically identifying value in the goals and items to be addressed in Product and Sprint Backlogs; **Flow** – for example by ensuring that value creating steps occur in tight sequence from backlog item to delivered Increment; **Pull** – for example through the pull of the most valuable items from backlog just in time to be be converted into valuable product

A common interpretation of previous versions of the Scrum Guide is that an Increment (often referred to as a *potentially shippable product increment*) is something delivered at the *end* of a Sprint after the inspection in the Sprint Review to check that it met its Definition of Done. On agreement that it is Done the Increment can then be released and its value realised.

Whilst Scrum still supports this mode of working, and many may choose to use it in this way, it does not fully embrace all the concepts of lean thinking – a concept that is given more support by a slight change in emphasis in the current Scrum Guide (Scrum Guide 2020).

The empirical pillars of Transparency, Inspection and Adaptation are embedded in lean thinking and Scrum theory as are concepts such as:

- Pulling work from a backlog – whether that is the whole Scrum Team pulling Product Backlog items into a Sprint or individual Developers pulling Sprint Backlog items to work on as part of the team practice of self-organisation
- Encouraging value through building in quality – Scrum does this by ensuring that the Scrum Team has all the capability needed to achieve Done and aligning Done with a *usable* Increment that delivers value.

Other features of lean thinking relate to value streams and flow. Scrum embraces the concept of having a defined and dedicated team focused on developing a single product and for work to flow through that team. The notion that an Increment is born as soon as a Backlog Item is Done clearly confronts the 'waste' in the system that comes from delaying delivery of value to the end of a Sprint in an Increment that incorporates multiple backlog items.

The Scrum guide clearly states that "*an Increment may be delivered to stakeholders prior to the end of the Sprint"* and *"The Sprint Review should never be considered a gate to releasing value*". This explicitly decouples the delivery of the Increment from the Sprint Review. Delivering value as early and as often as possible, and thus eliminating delays in realising value, is supported by the reality of delivering multiple Increments in a Sprint. The Sprint Review enables empiricism through inspecting "the sum of the Increments" delivered in the Sprint so that the learning from that can inform decisions about the next Sprint. Where needed by regulation or just custom and practice withing an organisation, the Sprint Review can be used to formally acknowledging what has been achieved – providing a regular 'accounting' checkpoint if that is considered valuable by the organisation

Performed correctly, the processes of Product Backlog creation and Refinement are very well supported by User Stories with each story having its own lifecycle in the context of a Scrum development. The Product Backlog is "an emergent, ordered list of what is needed to improve the product". When User Stories are created, they probably won't be fully formed, for example they may only contain a few words describing the 'I want…' and miss out on stating the ultimately valuable 'As a…' and 'So that…' statements. The concept of emergence supports the idea that the detail of Product Backlog items should emerge over time and empiricism suggests that 'the latest responsible moment' is the right time for each step of this incremental evolution to happen. Accordingly, as a Story gets closer to the top of the ordered list it well get increasingly detailed. It would be wasteful to explore full detail of all the stories in the backlog simply because there will be no certainty that it will ever get to the top of the emergent list. There is also a risk that detail explored too early will lose its validity over time as the product itself evolves and the goals are themselves refined to meet opportunities and challenges posed by the real world in which the product will exist. The last responsible moment for a story to be 'ready' for a Sprint (i.e. fully detailed in terms of As a…, I want…, So that… and fully formed through Conversation in terms of Confirmation and Context) is, therefore during the Sprint before the one in which it will be worked on.

Stories, Tasks, Progress and the Daily Scrum

During Sprint Planning, as a key part of topic 3 (How will the chosen work get done?), Developers often find it useful to identify the tasks needed to turn the User Story into an Increment.

For example, a User Story accepted into a Sprint with a Goal of "Agree specification for railway carriage retrofit" might be "*As a* train passenger, *I want* to be able to charge my portable electronic devices *so that* I can work or entertain myself throughout my journey". During Sprint planning this might break down into a number of tasks such as:

- Organize passenger focus group session to refine understanding of customer need
- Run focus group session and share refined understanding with entire Scrum Team
- Examine practicality of options based on needs identified from focus group
- Work with electrical safety expert to ensure we can provide power safely
- Estimate costs for 'gold', 'silver' and 'bronze' options for providing this facility

These tasks along with the User Story they relate to and the Sprint Goal make up the Sprint Backlog – the plan for the work needed created by the Developers for the Developers. Note that the Sprint Goal might emerge as a result of identifying the most important Product Backlog Items to bring into a Sprint or it might be defined in advance with appropriate backlog items selected to achieve the goal. Either way, the Sprint Goal, the commitment for the Sprint must be agreed as part of Sprint Planning.

The journey towards Done for a story and the work being undertaken to achieve that goal can be tracked on a physical or virtual KanBan board – a physical board recognised as being better for visualising progress for co-located teams and a virtual board recognised as being better for coordinating team activity for distributed teams.

In the case of a physical board, cards and/or sticky notes for tasks might be placed in a 'not started' column and when a team member selects a task to work on, the card might be moved to an 'in progress' column. On completion of the task would then be moved to a 'done' column. Note that the number of columns to indicate status and the labels used will depend on the nature of both the product being built and the development approach being followed.

If each team member has a 'row' on the board or if ownership of a card can be indicated in a different way the whole team and any other interested party will be able to see at a glance who is working on what story and the progress being made in the collective endeavour to achieve the Sprint Goal.

Physical boards or open access to view virtual boards provides *transparency*

The Daily Scrum should ideally be held in the physical or virtual vicinity of the KanBan board to provide a focus for the Developers as they *inspect* their progress towards the Sprint Goal and, with that in mind, *adapt* their plans for the upcoming day of work. The KanBan board should be updated during the Daily Scrum to maintain *transparency*

Many Scrum teams find charts such as burn-down charts useful in providing transparency of progress and to help focus their inspection of progress towards achieving Sprint Goals. Burn-down charts track 'estimated effort to complete' against time and are generally used within Sprints to visualise progress vs prediction on delivery of estimated Sprint Backlog items.

Burn-up charts, showing work completed over time and cumulative flow charts, showing the status of work items over time in term (e.g. in terms of not started, in progress and done) may also be useful. It is important to note that the main purpose of these charts is support

empiricism. i.e. to aid transparency, help focus inspection and in its next iteration to assess whether adaptation is effective. Because Scrum endeavours are not based on detailed up-front analysis and planning it is dangerous to use such charts as a mechanism for judging team performance.

Comparative Estimating

- **Comparative estimating** is:
  - Estimate of effort expressed as a multiple of that of a low effort, ideally known, job (something that has already been done) or a low effort job that is well understood
- The **rationale for using the Fibonacci-like series** of values
  - A non-linear scale is used because as estimates get larger, they typically get less precise because of an increase in uncertainty.
  - The Fibonacci sequence of numbers (1, 2, 3, 5, 8,13 ,21…) starts with 1 and 2, with each subsequent number being the sum of the previous two, so 3 (=2+1), 5 (=3+2), 8 (=5+3) etc.
  - The series typically used for estimating is only Fibonacci-like (0, ½ ,1, 2, 3, 5, 8, 13, 20, 40, 100, ∞)
    - 0 – no or trivial effort required (may com 'free' with another story)
    - ½, 1, 2, 3, 5, 8, 13, 20, 40 100 – estimable effort of increasing size
    - ∞ – effort >100 (too large to estimate, will first need to be broken down)

Planning Poker

The technique of **'Planning Poker'** for consensus-based, comparative estimating of User Stories that makes use of a special deck of playing cards. Each estimator has a set of cards with values like those described above. Smartphone and other Apps are also available. It is a serious 'gamified' technique that, unlike Poker, does not involve gambling!

- To start a poker planning session, the Scrum Team agree a previously estimated, low-effort, 'baseline' story with a 'story point' estimate of 2.
  If there is no pre-estimates story, the team selects one that, in their opinion, is well defined and of low effort and simply give it a value of 2.
- The Product Owner will select a Product Backlog item and describe it to the Scrum Team (the estimators)
- The estimators discuss the story, with input from the Product Owner as needed. When the feature has been fully discussed, each estimator privately selects one card to represent his or her estimate. All cards are then revealed at the same time.
- If all estimators select the same value, that becomes the estimate. If not, the estimators discuss their estimates. Estimators with the highest and lowest estimates typically share the rationale for their estimate. After further discussion, each estimator reselects an estimate card, and all cards are again revealed at the same time.
- The poker planning process is repeated until consensus is achieved, at which point the Product Owner selects the next story to be estimated

  Note: It can take a long time to achieve consensus around a single value so the Scrum Master – who acts as a moderator in the process – may suggest getting an estimate to within a range of 1 position on the scale and then 'call' the estimate. So, for example, if there are a range of 5s and 8s showing, it might be best to 'call it' as 8 and move on.

- Once the stories have been estimated, a second pass is recommended to recalibrate and/or bring learning from discussions of later stories to bear on those that were estimated earlier. In this second pass the team review all stories of similar value and seek

out any stories that seem mis-sized compared to the others of similar effort. E.g. they would look at all the stories estimated as '5' and ***consider*** changing its estimate to '3' if an individual story is noticeably smaller than the rest or changing to '8' if it noticeably larger.

Affinity Estimating

The technique of 'Affinity estimating' sometimes called 'bucket estimating' is a rapid variation on the planning poker technique that embraces both comparative and consensus ideals. In this technique"

- A work surface (physical or virtual) organised into columns (or buckets) with each labelled with a story points value from the Fibonacci-like sequence above
- As with planning poker a story is identified as '2' and placed in that column
- There are then three variants of the process
    - Variant 1:
        - The estimators swarm around the work surface and individually take stories from the backlog and, based on their opinion alone, place it in the column of the estimate they think it is closest to.
        - Once the backlog has been exhausted the estimators spend a small number of minutes moving cards around that, based on their opinion alone, have been incorrectly placed.
        - The Scrum Master – who again acts as moderator – identifies any stories that seem to be moving around a lot, perhaps pulling them from the work surface or pinning them in place.
        - These stories are then discussed collectively – with input from the Product Owner – to gain consensus on the final estimate. As with the planning poker technique, the higher of two entrenched opinions should normally be taken
    - Variant 2:
        - The Product Owner reads the next story
        - Any estimator can ask a clarifying question and after hearing the answer indicates a column for the story to be placed in, asking for acceptance from the group. If there are no questions any estimator may indicate a column
        - If there is disagreement the story is discussed further – ideally prompted by a question from another estimator. This step is timeboxed to an agreed duration e.g. 60 seconds.
        - If disagreement remains, the story is put to one side and the process is repeated for the next story
        - After the remaining stories have been estimated, the stories that were put to one side are discussed until consensus is reached. As above, the higher of two entrenched opinions should normally be taken
    - Variant 3 – in which Estimators take it in turns to propose an estimate
        - An estimator reads the first story and proposes an estimate
        - If there is any dissent the story is discussed until consensus is reached
        - The next estimator in turn reads the next story and proposes an estimate and so on…

Commitment Based Planning

Commitment Based Planning – A simple technique that can be facilitated by the Scrum Master to help the Developers achieve Topic 2 of Sprint Planning

- From the Product Backlog, ordered by the Product Owner, the Scrum Master selects the first Product Backlog item.
- S/he then poses the question to the Developers: "If this was the only item that we needed to get done in the Sprint, could you **commit** to getting it 'Done'?"
- If the Developers say "Yes we are happy to commit to that" then the Scrum Master selects the next item from the ordered list and poses the question: "If this story and the previously committed item(s) were all we needed to get done in the Sprint, could you **commit** to getting it 'Done'?"
- If the answer to that question is "Yes" then the previous step is repeated until the answer is not a firm "Yes"
- If the answer is "No" or even "Not sure…", then a short discussion can be held as to whether there are any other stories near the top of the ordered list that could be included in the commitment.

Note: When combined with the comparative estimating technique described above, this technique can be used in the first Sprint to *predict* a Velocity for the Scrum Team.

- Velocity is the number of story points associated with 'Done' User Stories from the previous Sprint.
    - In some cases, Scrum Teams might prefer to take a rolling average of points from a number of previous Sprints.
- Even where there is an established Velocity, this technique may be used in the event of change of team members or any other change that might impact team performance
    - Otherwise past performance as demonstrated by an established velocity is a better predictor of future performance
- If Story Points and Velocity are not being used then Commitment Based Planning is a good default technique to address Sprint Planning Topic 2