

Delivery Techniques

- User Story Mapping
- 3Cs (Backlog Refinement)
- User Story Writing
- Cost of Delay Prioritisation
- Effort/Value Matrix (2*2) Prioritisation
- BDD - Behaviour Driven Development
- Vertical Story Slicing
- User Story Splitting
- Estimation - Planning Poker
- Sprint Planning
- Sprint Retrospectives
- Sprint Review
- Infrastructure as Code (IAC)

User Story Mapping

□ What is it

- User story mapping is a visual representation of the user journey across a product or service. It helps teams understand the functionality and prioritise development tasks, ensuring that the user's perspective and needs remain at the forefront.

□□ Who

- User story mapping only delivers any benefit when done as a team activity. The whole core team should take part in this activity to ensure diversity and aligned understanding on any MVP candidates.

□□ Running the technique

- Find a large wall or digital collaboration tool like Miro. Begin by outlining the backbone of your user's journey. This is a series of high-level activities or key steps the user goes through when interacting with the service.
- These are positioned sequentially from left to right, reflecting the order in which they typically occur.
- Below each high-level activity, you'll break it down into more granular tasks or sub-activities.
- Now, for each of these tasks or sub-activities, you'll develop a user story or backlog item.
- The vertical positioning of user stories under each activity can indicate priority, with higher priority items at the top.
- This helps teams understand what should be developed first to deliver the most valuable features to the user early on.
- By viewing the map, teams can group certain user stories together into logical releases or iterations. This is especially useful when determining the features for a Minimum Viable Product (MVP) versus those for subsequent releases.

□□ Additional Material

- <https://jpattonassociates.com/the-new-backlog/>

3Cs (Backlog Refinement)

□ What is it

- The Three Cs stand for Card, Conversation, and Confirmation and is a approach to refinement of the backlog items so that they are ready to be considered for inclusion in a sprint backlog.
- It's a technique that emphasises the importance of a user story (Card), the discussion around it (Conversation), and the acceptance criteria (Confirmation) to ensure clear understanding and expectations across the team tasked with delivering that value.
- A “Card” (or often a Post-It note), a physical token giving tangible and durable form to what would otherwise only be an abstraction:
- A “Conversation” taking place at different time and places between the various people concerned by a given feature of a software product: customers, users, developers, testers; this conversation is largely verbal but most often supplemented by documentation
- The “confirmation”, finally, the more formal the better, that the objectives the conversation revolved around have been reached.

□□ Who

- Primarily lead by the product manager

□□ Running the technique

- A Card (or often a physical or virtual Post-It note) is just a placeholder that signifies there is a customer requirement. It's like an item on a to-do list or a reminder taped to the fridge. A token. Create a user story with a title and value statement and maybe one or two acceptance criteria
- Next engage your team members or a service owner in a Conversation. Embellish and refine the first draft from insights gained from this Conversation. This should include a description and further acceptance criteria.
- Ensure Confirmation with a final refinement session that validates mutual understanding of the valuable outcome being targeted, what requirements need to be satisfied, and an estimate of the effort to achieve this

☐☐Additional Material

- <https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>

User Story Writing

□ What is it

- User story writing captures product requirements from the perspective of the end user that derives a valuable outcome from this requirement being delivered.
- It must have a short meaningful title, value statement, and acceptance criteria, but may also have a description or other supporting business and technical insights that helps the team to estimate and deliver the requirements.
- Apply INVEST principle to improve focus on value outcomes

□□ Who

- User story writing is undertaken primarily by someone acting in the role of product manager. During and post refinement other team members may contribute to the user story.

□□ Running the technique

- The user story title should reflect the valuable outcome, and not the intermediate steps. Try to use a single active verb and object. e.g. Display current personal details
- The value statement should take the form of ... AS A (describe the end user who has a need) I WANT TO (the action to take) SO THAT (the benefit/ valuable outcome achieved)
- Write a description of what goods looks like when this user story is delivered, if this may help refinement with team members
- Ensure the user story adheres to the INVEST principle: Independent, Negotiable, Valuable, Estimable, Small, Testable
- Draft the acceptance criteria that needs to be satisfied. Please refer to the BDD technique guide for further detail.
- Engage with relevant team members to refine and estimate.

□□ Additional Material

- <https://www.knowledgehut.com/blog/agile/powerful-tips-for-writing-the-best-user-stories-in-scrum>

Cost of Delay Prioritisation

(also known as CD3 or WSJF)

□ What is it

- A prioritisation approach to make value and urgency more visible. It quantifies an opportunity cost in economic terms when work is delayed, enabling more objective decision making and trade off decisions.
- There are two basic inputs required to work out the Cost of Delay:
 - 1. Value – Calculate a value of the benefits per period, based on an estimation of the work's economic value to the organisation over a given time period.
 - 2. Urgency – An understanding of the work's urgency: when do the benefits start to occur or deteriorate? Is there a calendar peak period that is critical to make ?

□□ Who

- The technique is typically used by a product manager as part of prioritising their product backlog and gaining consensus across stakeholders with conflicting viewpoints of what work is most valuable

□□ Running the technique

- Calculate a benefits figure for the change derived from value definition work already undertaken.
- Work out when the benefits would start being realised or when the expected benefits may deteriorate if at all
- Derive a Cost of Delay figure using a standard period of time e.g. month
- Repeat the steps for initiatives or backlog items competing for the same capacity
- This provides an objective comparison on which to achieve transparent consensus

□□ Authoritative Source

- Principles of Product Flow (Donald G. Reinertsen)

Effort/Value Matrix (2*2)

Prioritisation

□ What is it

- It is used to help quickly prioritize a backlog of work associated to a service or product based on relative prioritisation. Each of the items are assessed against two criteria: the value that is realised from delivering it, and the effort required to complete it.
- It is the recommended prioritisation approach for backlogs with a large number of items(>20) for teams with little previous experience or competency in this activity.
- The key benefits for this technique are: Quick to understand and use for team members new to the service/initiative/product; strong visualisation to engender further discussion; highly productive in processing large backlog volumes in a short space of time; inherently enables high granularity of ordering; clear recommended next actions for each quadrant

□□ Who

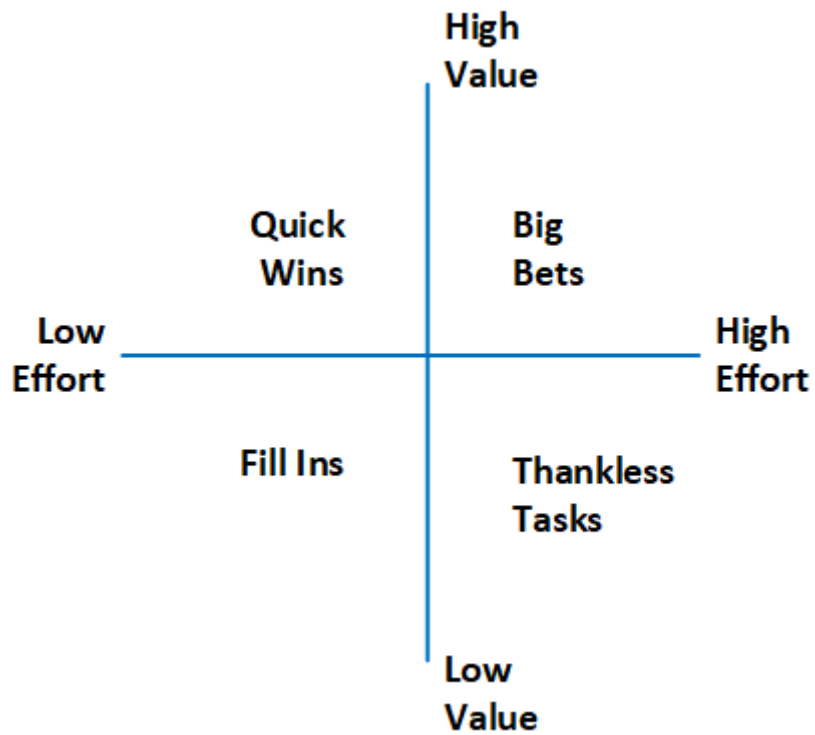
- Lead by the product manager, this technique only works well with a cross functional team and service SMEs working in a collaborative environment, with presence of at least one architect or developer

□□ Running the technique

- Find a backlog item that to use as the anchor: find a previously delivered item on the backlog with a clear purpose, and effort taken and value expected would roughly enable it to sit in the middle of the matrix.
- Agree as a team to how value is defined to ensure consistency throughout the exercise
- Assess one item at a time, one person facilitates selecting and sharing details. Discuss and agree where to place it relative to the anchor item.
- Start with two or three more straightforward items (helps build confidence and understanding)
- Once completed, next recommend actions are Big bets (top right) should be further broken down to identify and derisk the large effort estimate. Quick wins (top left) should

be grouped against top level objectives and requalified.

- Agree sequencing of the items in the top left and top right segments.



BDD - Behaviour Driven Development

□ What is it

- Acceptance criteria in the context of BDD refer to the specific behaviours a system should exhibit in order to be considered correct, the scenarios in which all possible acceptable behaviours are described. The approach focuses using plain language that everyone involved can understand.

□□ Who

- The primary creators of acceptance criteria is the product manager
- During refinement the product manager works collaboratively with the tester and developer to review and baseline the acceptance criteria and efficacy of the syntax

□□ Running the technique

- Identify each possible scenario that might validly occur for the backlog item
- For each scenario, define the behaviour that must be met for it to be considered done and accepted
- Define each behaviour in the form of : GIVEN... (define the precondition, setting the scene) , WHEN.. (define the action to be taken, the thing we are testing), THEN...(define the expected result occurring from that action). An 'AND' maybe added to any of the three statements.

□□ Additional Material

- <https://openpracticelibrary.com/practice/behavior-driven-development/>

Vertical Story Slicing

□ What is it

- Vertical story slicing" is a concept used primarily in agile software development, specifically in the context of breaking down user stories. The main goal of vertical slicing is to deliver a thin, yet functional slice of the overall system that cuts through all the architectural layers and components, delivering value to the end user. This contrasts with "horizontal slicing," where you might build out a complete layer of the system, but the user sees no functional value until all layers are completed.

□□ Who

- A collaborative activity involving the product manager, developer and architect

□□ Running the technique

- Functional Value: Each vertical slice should provide some kind of value to the user, even if it's minimal. This means that after implementing a vertical slice, there should be a usable piece of the product, even if it's only a small feature.
- All Layers: A vertical slice touches every layer of the application. If you imagine the system as a layered cake with a UI layer, service layer, and database layer, a vertical slice would be a thin piece that cuts through all those layers.
- Small and Manageable: Each slice should be small enough to be developed within one iteration. This allows for faster feedback and quicker adjustments based on real-world use or stakeholder feedback.
- Minimize Dependencies: By focusing on vertical slices, teams can minimize dependencies on other parts of the system. This reduces waiting times and increases the team's ability to deliver continuously.

User Story Splitting

□ What is it

- User story splitting breaks down a large user story into smaller multiple user stories that can be delivered earlier given their reduced complexity and size. Smaller sized user stories are easier to accommodate into the overall sprint plan.
- This technique is best employed on user stories that have a large effort estimate.
- The approach to splitting the user story should ensure value can be delivered to the user in each smaller story, so still adheres to the INVEST principle

□□ Who

- A collaborative activity with the product manager and developers involved with the original user story

□□ Running the technique

- Find the best pattern to split the story. The functionality associated to the story typically leans towards one option more than others.
- Tried and tested patterns are : Workflow Steps, Operations to perform, Business rules that need to be applied, Variations in data, User interface options, simple functionality/complex functionality, defer performance
- After applying a split test each candidate against the INVEST principle. If one of the smaller stories fail, find another user story that it can be merged with.
- The number of split stories created will depend on the scope and complexity of the original user story. Use a guideline that each split user story should take no more than 15% of your sprint capacity to achieve
- Reprioritise the split user stories - you may find that some of the smaller user stories are not as valuable as others.
- Please refer to the additional material source for the full flowchart of splitting patterns

□□ Additional Material

- <https://www.humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories/#flowchart>

Estimation - Planning Poker

□ What is it

- It is a consensus-based estimation technique that produces quick and reliable estimates backed by the whole development team. It is highly recommended to purchase a purposed planning poker card deck.
- A good planning poker card deck is based on the Fibonacci series- 1,2,3,5,8,13,20,40,100

□□ Who

- Everyone that does the work is involved, so typically only the developers should be involved
- If the work item is wholly related to a specialism, e.g. research, UI design, testing then only team members who have experience of that specialism should take part.

□□ Running the technique

- Agree consensus on average value using a previously worked backlog item, e.g. a value of 5
- The backlog item is presented by the product manager or delivery manager, with a short team discussion
- Each team member allowed to vote, does so privately. The cards are then all shown.
- If the scores are in general consensus, e.g. 3, 2,5, 3, 3, then the score is recorded
- Where there is notable variation, e.g. 2, 3,3,5,10,5, then the members with the highest and lowest scores share their thinking. Everyone then repeats the activity, until consensus is reached.

□□ Additional Material

- <https://www.mountangoatsoftware.com/agile/planning-poker>

Sprint Planning

□ What is it

- It's the event where the team determines the backlog items they will work on during that sprint and discusses their initial plan for completing those backlog items. Typical duration should be between 1 and 2 hours.
- The sprint planning session should not involve any significant focus on refining or estimating backlog items. These activities should have been completed by the team in advance of sprint planning, and only the odd exceptional item should be raised and discussed during this session .

□□ Who

- Facilitated by the delivery manager, this technique involves the whole core team

□□ Running the technique

- The product manager should begin the sprint planning session with proposing the draft sprint goal(s), and explain the rationale behind them
- The team discusses and agrees with the sprint goal.
- The product manager then proposes their candidates for inclusion in the sprint, with particular focus on the value being delivered by each item.
- Each candidate item is summarily presented with the opportunity for any of the team to raise any last minute questions or concerns on aspects such as estimate, acceptance criteria or the proposed value outcome
- The delivery manager then facilitates discussion with the team regarding the confidence of achieving all the scope against expected capacity for the sprint. This may result in negotiating with the product manager to reduce the scope of the sprint backlog.

□□ Additional Material

(Agile bootcamp handbook.)

Sprint Retrospectives

□ What is it

- Regular Retrospectives are dedicated sessions where teams reflect on their processes and performance, aiming to identify areas for improvement and celebrate successes.
- The event is a safe place with no observers or participants outside of the core team, encouraging an open and transparent dialogue within the team

□□ Who

- Facilitated by the delivery manager, this technique involves the whole core team

□□ Running the technique

- Retrospectives should be scheduled on a regular basis at the end of a sprint or iteration for around a one hour timeframe
- The team should jointly agree the ground rules of running this event
- Discussion is typically facilitated by the delivery manager, although volunteering to run the session offers learning opportunities in more established teams.
- Discussion should focus on how work is done and the dynamics and working relationships within team members. Discussion on the content of the work completed should be excluded- this is covered in a sprint review event
- All voices and viewpoints should be treated as equal and valid in this session, irrespective of the role
- It is recommended to use a template to help frame the conversation. There are numerous freely available templates that can be used in a physical or virtual meeting.
- Good sessions should surface at least one proposed improvement to take forward and implement in the next iteration

□□ Additional Material

(Agile bootcamp handbook)

Sprint Review

□ What is it

- Sprint reviews takes place in an informal atmosphere at the end of each sprint, with the team sharing what has been achieved with stakeholders and where possible end users. The review provides a valuable opportunity for early feedback on the deliverable increment from the people that matters most- end users and stakeholders. Sprint reviews are also known as Show and Ask intimating the conversational nature of a good sprint review event.

□□ Who

- Product Owner leads the sprint review, but all the team should attend as this is an opportunity to hear feedback first hand

□□ Running the technique

- Plan for about 60 to 90 minutes to run the review for a typical 2 week sprint.
- It is recommended that the team employ an open invite approach casting a wide net across interested communities to maximise feedback on the deliverable increment.
- Open the sprint review with a reminder of the sprint goal, and then present in an informal manner the deliverables completed during the sprint. It is good practice to encourage the developer/designer of a specific deliverable to present this to the audience.
- Encourage open dialogue and feedback from the audience, and note down any specific points of interest that might be worth further exploration in a future sprint, and refinement of the existing backlog.
- Complete the review with a quick summary outlook of the product backlog and likely candidates for the next sprint.
- Following the sprint review, the team discusses any new insights from the feedback and amend the product backlog if appropriate.

□□ Additional Material

Infrastructure as Code (IAC)

□ What is it

- Infrastructure as code (IAC) is a formalised way of scripting infrastructure requirements that can then be played, and replayed, in a target environment.
- Because the setup is scripted it removes the possibility of human error when executing and can be backed up and managed in source control.

□□ Who

- DevOps engineers.

□□ Running the technique

- There are different formats required for different targets. Azure uses ARM, for example, Terraform can be configured cross-cloud, Vagrant can configure virtual machines and Kubernetes manifests are also in this category.
- Just like with software, you should test your IaC on a dummy environment before making changes to anything live.
- Your deployment pipeline should run the scripts in a target environment as part of the deployment process: Build, Test, Create (Iac), Deploy

□□ Additional Material

- <https://www.bmc.com/blogs/infrastructure-as-code/>