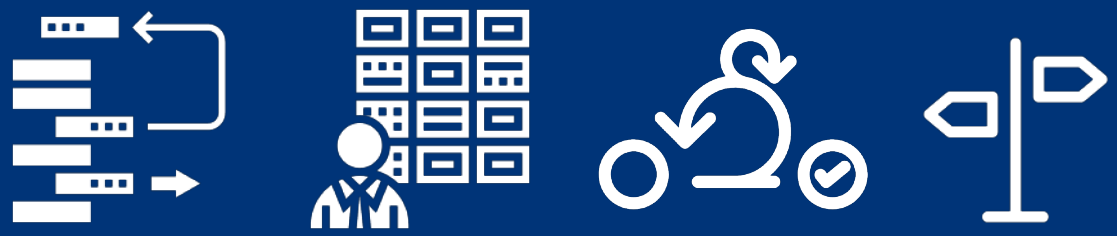
Contribyte

Backlog Management Guide

How to do effective backlog management. A guide for Product Owners.



Introduction

What is this guide?

This guide is targeted to help Product Owners tackle one of the most important responsibilities that they have: how to manage the product backlog. The guide contains work methods, tools, organization methods and things to take into consideration when working with the backlog.

Product Owner is the one person who makes decisions on priority and description of the work items that get developed for the product. In smaller R&D organizations, this role is easy to identify and define. In larger organizations, where multiple teams are cooperating to develop one or more products, the task of deciding priorities and working on backlog items becomes quickly too much to handle for one single person. In these situations, multiple people work together. Even then, one of these people is the main Product Owner, while others assist him or her.

Even in these cases, knowledge of the backlog management tasks, tools and methods described in this guide can help remind everyone to spend enough time and energy defining what gets built.

Who is if for?

This guide is meant for people who have the title or role of Product Owner. If you set the priorities and are responsible for the work item specification for a team or a product, you are either a Product Owner, or assisting Product Owner.

Remember, sometimes you do not have a team – many development activities use subcontractors to deliver the implementation, while a person on the customer side actually helps to define and prioritize what gets built. In this case, the customer representative is a Product Owner. This guide is also useful for him.

Backlog is important – it is the only input the team gets on what and why. Improving the backlog and management of the backlog is one of the most important things for Product Owners.

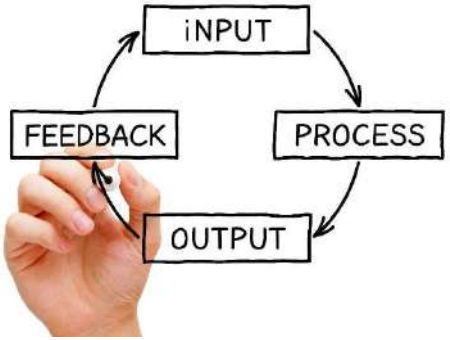
Backlog Quality and Team Motivation

Backlog quality is a function of

- How items are approved into the backlog (funnel management)
- How items on the backlog are prioritized
- How items on the backlog are scanned and studied
- How items on the backlog are refined

The quality of the backlog is important, as backlog items are the only input for team process. If team gets poor quality backlog items that arent prioritized correctly, it can never maximise the results.

Backlog quality is the foundation for a successful team. All successful teams invest time on making the backlog handling ceremonies better.



Product Owner should invest enough time on backlog management (minimum 4 hours a week / 25% of time)

The best way to do refinement is together with the team, however this does not mean that Product Owner cannot prepare the epics and stories beforehand.

Product owner can also enlist the team's help to prepare backlog items for refinement. Preparation work speeds up the refinement meeting, but the discussion in the actual meeting is still invaluable.

Team must invest time on refinement and associated feasibility studies and spikes

Good guidelines to start from

- Refinement session regularly, once a week, 1 hour, full team
- Regular refinement session aims to keep top of the backlog 1.5x – 2x velocity refined
- Regular refinement session scans the rest of the backlog for trash cleanup, reprioritization needs and feasibility study needs
- Team and PO prepare to refinement sessions separately
- Feasibility studies and spikes are taken into sprint backlog when needed

Common Challenges With Backlogs

Too long backlog

Backlog is not transparent and public

Backlog mud

Team is reluctant to invest time to refinement –too PO centric

Product Owner too busy to make backlog better

Tools of Backlog management not organized

Overspecification

Backlog items unclear (one liners)

Leaks – team gets work from many sources

Backlog too long too long backlog leads to inefficient work and waste, and increases uncertainty

Backlog not transparent hiding the backlog decreases motivation to manage and refine backlog items

Backlog mud allowing backlog to contain items that will never get done and are decomposing will demotivate everyone from touching the backlog

Team reluctant to invest time on refinement team and PO have to collaborate on refinement. Team has to be made see the value of spending time on refinement

Product Owner too busy a busy PO will not manage backlog effectively, and leads to team spending time on wrong things or things wrongly specified

Tools for managing backlog not organized working with linear backlogs is not efficient

Item overspecification limits agility and self-management of team and decreases motivation for refinement

Items unclear (one liners) unclear one liners will cause overengineering, sprint spillover and uncertainty in release estimates

Process leaks if backlog management is not working, team will encounter work leaks from other sources

Backlog Size and Tips On Size Management

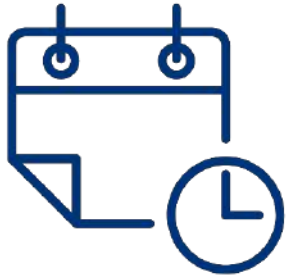
Long backlog makes backlog management inefficient and demotivates everyone involved in it's refinement. It makes communication of things added to backlog unclear, and increases risk for important things getting lost among things that are slowly decomposing at the bottom (backlog mud)

Keep the backlog under 150 items long.

Keep the backlog under 6 months of work for the team(s).

See backlog organization and funnel visualization for more ideas how to achieve this.

Number of items ~<150
(per PO?)



Length of work
(2-6 months)

Do not allow backlog to grow too long.

Too long backlog will

- be more difficult to manage
- decrease motivation to refine it
- increase risk of surprises and "backlog mud" on the bottom of backlog (things that practically never get done)
- increase likelyhood for your adding things to the backlog (because it already is too long, and contains all wishlist items, why not add some more)

Size management tips

- Auto kill old items
- Active decisions and communication
- Design In Progress limits (similar to work in progress, WIP) for different backlog funnel statuses
- Clear strategy and process for backlog management

Backlog Communication and Tips

Product Owner is responsible to keep the stakeholders of the product well informed on the backlog and items that interest specific stakeholders.

Manage expectations – unless you tell them otherwise, the people who requested will assume that their request is on the top of your backlog, and implemented practically instantly.

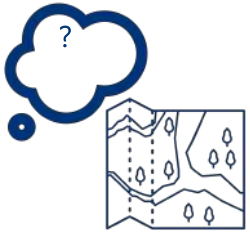
"Buyer mentality" – the people who request think they have to get their request from you as soon as possible. Manage this by showing them the full backlog and your reasons for prioritization as you have decided.

Demo effectively to many

Share progress effectively

Chat 1-on-1 regularly

1-on-1: update with progress and Data presents



What does each stakeholder want?

Good practices

- Mark originator to all tickets lest you forget
- Never allow "one liner requests" – book a time with the requester to commonly edit a description
- Be clear in your communication – do not answer "I will add it to backlog" when you only plan to add it to wishlist, or have no idea on the position it will have on backlog.
- Communicate your full backlog transparently to all requesters
- Communicate their request position on the backlog
- Use prototypes to verify what you plan to deliver
- Use online order analogy
  - Mandatory fields when ordering
  - Order confirmation
  - Inventory levels
  - Delivery date estimate
  - Order status regular email note
  - Shipment notification
  - Pickup notification

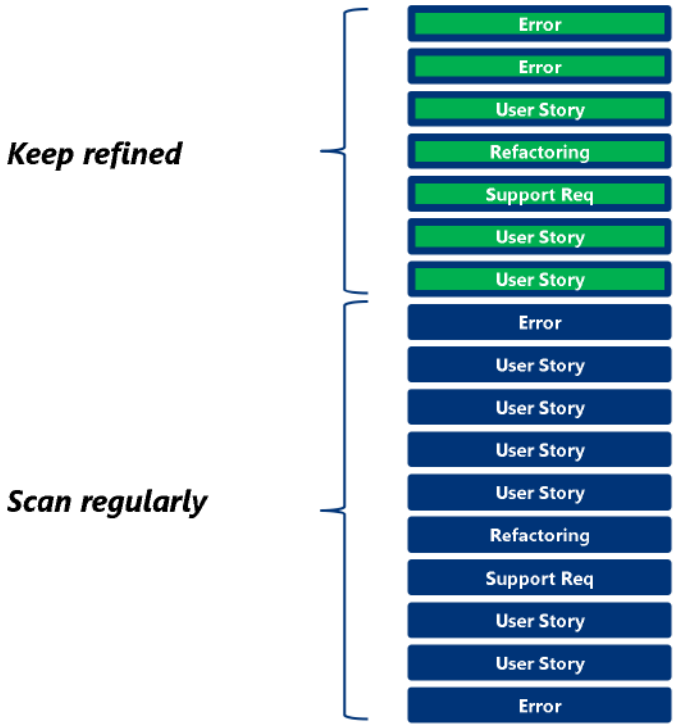
Focus, Refine and Scan

Keep top of the backlog refined – at least 1 x sprint velocity, preferably 1,5 – 2x velocity.

Trying to refine user stories beyond 2 x velocity leads often to waste.

In addition to refining the top, you should regularly scan the complete backlog for items that

- May require reprioritization
- May require feasibility studies or similar pre-work
- Have become trash, backlog mud or require cleanup



- **Keep regular refinements** often and long enough to maintain 1,5-2x velocity of backlog in refined state
- **Have a status, label or similar to indicate refined items**
- **Have Definition of Ready written and agreed** with team to indicate criteria for "refinemed passed"
- **Spend monthly time** on scanning the full backlog. This time can also be used to do "release planning" which can plan further than a couple of sprints
- **Spikes or feasibility studies** should be done for items lower down on the backlog or upcoming unclear large items on the funnel.

Funnel Visualization

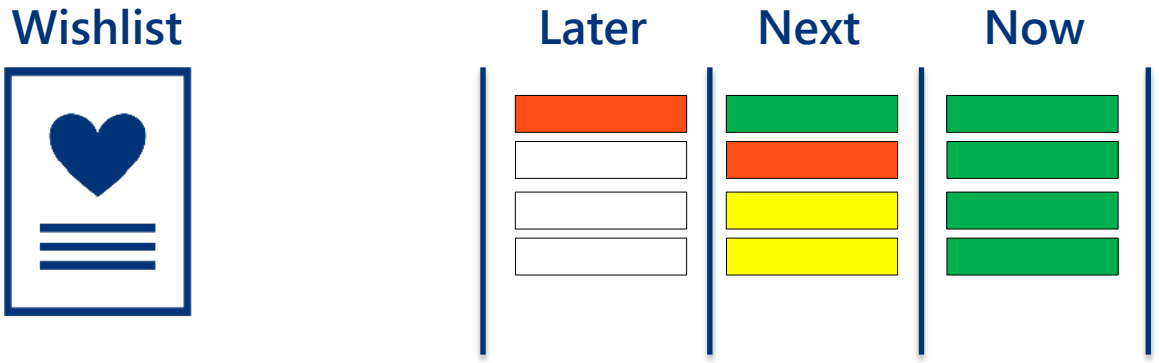
Now, Next, Later

Instead of a top to bottom "backlog list" – organize your backlog into a funnel. A good visualization of the funnel is a "Now, Next, Later" column view of the backlog. The backlog is then prioritized in these three parts – Now is prioritized separately, and Next and Later columns as well.

This has the added benefit of communicating to stakeholders also the level of understanding and uncertainty of the features to be implemented – you can educate your stakeholders to understand that items in "Now" are well understood and specified, and their size is accurately known. Therefore, any schedule commitments on these items are "holding".

In contrast, items in "next" are usually in the process of being understood. Their technical complexity is under study. Commitments are "target" rather than hard promises.

Items in "later" have not been investigated at all, and should a need arise to make commitments to them (i.e. sales case that requires a hard deadline) they should immediately be promoted to "next" for study.



Now next later planning horizon is typically 3-6 months

Funnel visualization methods

Now, Next, Later Organizing your committed backlog into columns of "Now, Next, Later" will give following benefits

- Allow you to communicate understanding maturity and level of uncertainty to stakeholders
- Allow team to understand which items need to be studied
- Allow team to balance effort invested to Now vs Next
- Allow prioritization of items in Next and Later to ensure highest importance items progress first

Wishlist Keep wishlist items separate from "committed" backlog

- when you hear, encounter or get requested an item that you immediately cannot be certain should be on the backlog, you can place it to the wishlist
- You should scan through and be aware of the wishlist contents and actively listen to feedback from stakeholders if any items on the wishlist should be promoted to full backlog status
- Remember – almost all the main backlog items are close to equal value. Wishlist items are of less value or uncertain value. These items should be actively tested.

Backlog Organization Methods

Linear Backlogs Work Only At Start

Linear backlog means a backlog that uses nothing to "group" backlog items under some umbrella. This is potentially useful at project start, when the backlog is empty and has to be grown fast. But as soon as the backlog grows to be bigger, it starts to be difficult to manage and prioritize.

One of the key things a product owner can do to make backlog management easier is to use some method to organize backlog. This means in principle that items on the backlog are collected under a topic, theme, functionality area or an epic.

Prioritizing under an umbrella becomes much easier, when you have just a few items to prioritize compared to hundreds. The next question is of course to prioritize different umbrella areas, but this is also usually easier than a single, linear backlog of hundreds of items.

Epics

Themes

Mother features

Technical debt

Categories/classes

Components

Use any of the following to "divide and conquer" backlog and make management easier

EPICS Large containers of stories, typically "mini feature projects" that have a lifetime of 2-12 months

"Mother features" (feature folders) Similar to epics, but more permanent. Feature folders are areas of features or functionality, that can be used to group features into a larger whole. More difficult to prioritize than epics.

Categories / classes Types of backlog items such as user experience, enhancement, robustness, performance..

Themes Larger than epics, themes can span multiple products and multiple backlogs throughout the organization

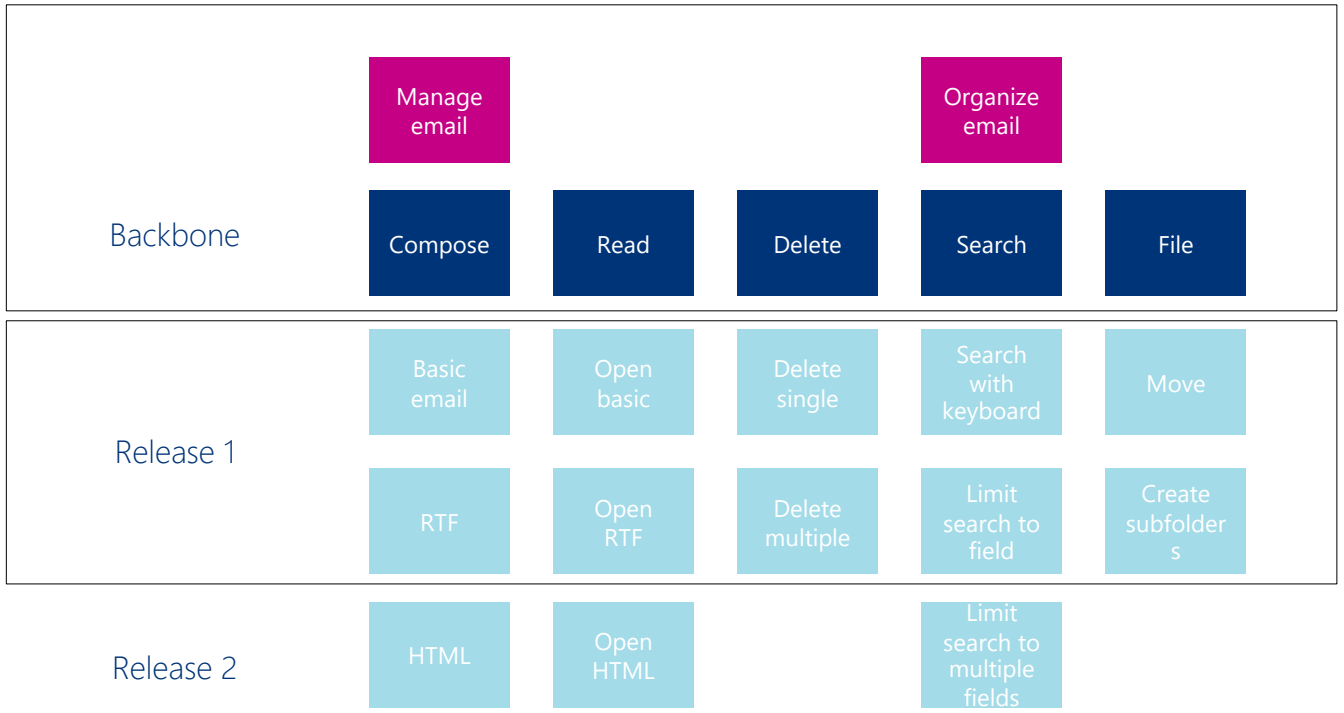
Components What components are used to build the solution? The components can be a powerful backlog management tool, especially if your organization is split into component teams

Technical debt Grouping technical debt tickets separately allows the team to monitor the level of debt (note – a growing trend on tech debt can also mean hidden technical debt is identified,not that the debt is necessarily increasing as indicated)

Story Mapping

Story mapping is a great tool

Story mapping helps teams and Product Owners organize the backlog. In story mapping, the features and functionality of the product is mapped under a backbone, that offers two levels of hierarchy above the individual backlog items.



Use story mapping technique to find items to organize backlog under

Backbone items are usually feature folders or epics

Agreed releases can be used to organize backlog based on targeted releases

Build story map based on a user journey
For example – user journey might be

- Log in
- Create new invoice
- Enter data
- Attach receipts
- Review
- Submit
- Check status

The above become epics, and contain detailed stories under them. Simple stories are positioned to the top, building the simple and most important functionality first. More complexity is added with new stories below the first one.

The more important features are always nearer the top. Find importance by comparing items to each other and moving the "winner" upwards

The items on the story map are usually only on title level – they have to be further refined by adding description and acceptance criteria to them before implementation

Backlog Metrics and Dashboard

Dashboard is a view that you use several times a day to monitor new items, progress of currently ongoing work, and can use as a report of status of work to stakeholders. A good dashboard contains lists of work and charts that give information on the project status.

NEW DEVELOPMENT

Dashboards on new development work will allow you to share progress and predict outcomes of releases. Teach your steering stakeholders to read the dashboards directly to make it easy for you to communicate status.

BUGS

Bug lists allow you and the team to react quickly and correctly to any new events, such as new bugs, or updates to urgent and critical bugs. Metrics allows you and the team to understand if your situation is getting better or worse.

Good backlog dashboard items:

- Release burn up chart
- Sprint burn down chart
- Continuous Flow Diagram
- Last updated item list
- Assigned to me
- Team sprint velocity history

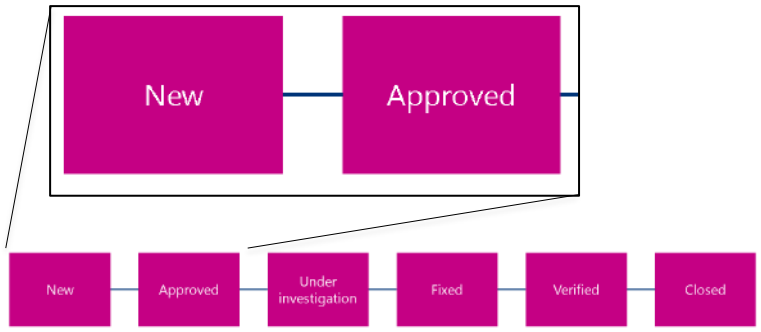
Good bug dashboard lists:

- List of new bugs (latest or in new state)
- All open bugs listed in order of severity, with information who is handling them and when is the latest change or comment done
- Your own bugs (in case one ping pongs back to you for comment)

Good bug metrics:

- Bug average age
- History of open bugs (open bugs per date)
- Bug distribution (pie chart or similar)

Good Quality Bugs



Information	Description
Title	A short and clear title.
Severity	Severity (fatal/showstopper, critical, major, minor).
Tested on	Information on what version it was tested on (Software under Test, SUT).
Environment	Information on the environment where it was tested.
Preconditions	Preconditions for the reproduction steps.
Steps to reproduce	Reproduction steps with which the bug can be reproduced.
Expected result	What was the expected result based on specification or tester intuition.
Actual result	What was the actual result.
More information	More information on the error:
When	When it was first noticed.
Where	Where it was first noticed (environment, version).
Recovery information	How the end user can recover from the error situation.
Reproducibility	How often the error is reproduced (every time, easily, hard to reproduce, cannot reproduce).
When not	When is the error not encountered (different execution steps, different environment, time of day, etc.).
Originator	Originator (who first reported the error, if it's not the person creating the error ticket, with contact information)

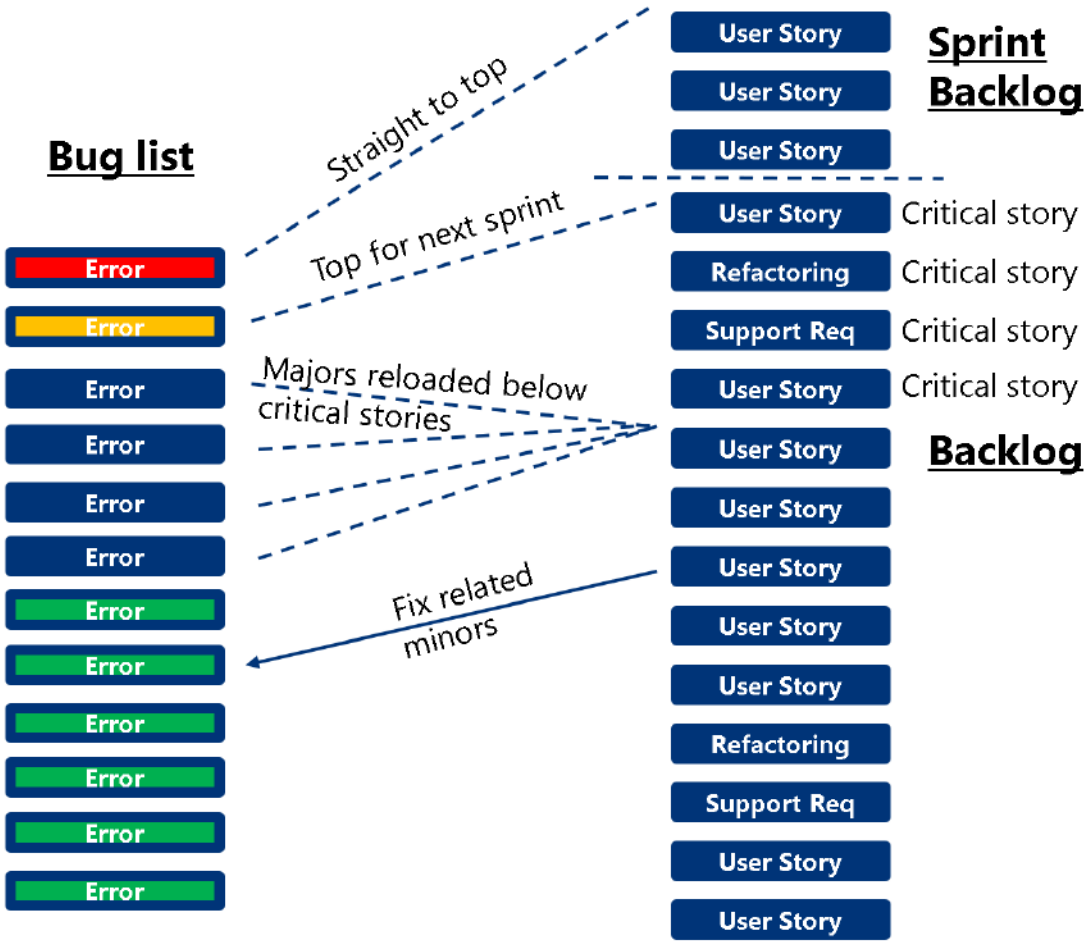
Bugs need quality control too, same as backlog items. The team should discuss what a good bug looks like. See table for example of good quality bug info fields.

Filling all the info fields does not take that much time, but it makes finding the root cause and fixing the bug much faster.

Bugs should be screened daily to check that they are of good quality, all necessary info fields are filled, the severity/priority matches the description. One further important thing to check is the reproduceability.

A good practice is for the Product Owner to do the bug screening. He can set up a workflow to separate new bugs from ones that have already been screened.

Bugs As Part of Backlog



Errors are part of the backlog. However, you can decide if you want to keep error list separate from main backlog. A separate error list makes sense if your team has made clear "contract" how different errors are treated. Such contract can say how bugs of different priority are located in relationship with rest of the work on the backlog.

For example

- fatal errors are always treated instantly upon detection
- Critical errors always are started next or make it to next sprint automatically
- Major errors are always on top of backlog, unless a backlog has specifically agreed "urgent" items
- Minor errors are treated separately, and get bandwidth only as their number is sufficient, or if team otherwise touches areas of code where they are lurking

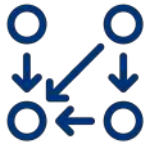
This contract should be made with the team, and for this to work properly, the bug priorities need to be screened and decided actively

Backlog Management in Large Organizations

Common backlog management challenges in large organizations:

Lack of sufficient analysis

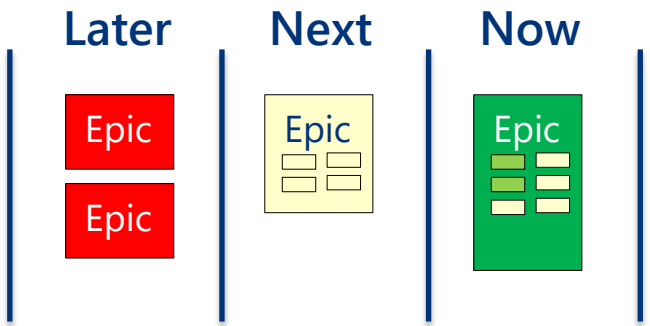
Business does not prioritize



Inadequate dependency analysis

Scaled agile methods (SAFe, LeSS, Scrum@scale, others) aim to solve these challenges by bringing in the necessary people and roles to plan, refine and prioritize together.

Rolling planning models such as Now, Next, Later or Boulders, Stones, Pebbles will further attempt to visualize and guide the organization to add clarity to backlog items as they get nearer to implementation in the product backlog funnel.



Backlog Management in large organizations requires business people to participate and prioritize.

Prioritization is not possible in large organizations without it happening in high level (epic level or theme level). Individual backlog items cannot be prioritized by themselves. They must inherit the priority from "above".

Business cases are used on project or initiative level to first find out project level priority decisions.

After this, projects prioritize the epics, and this priority decision is then inherited into the actual backlog items

Teams analyse and split epics into backlog items. This can be done with story mapping or other story splitting methods.

ADVERTISEMENT: THIS IS WHAT YOU NEED. TALK TO YOUR BOSS OR WHOEVER HAS MONEY. EMAIL US. RIGHT NOW.

Become World Class

Improving your skills as Product Owner benefits the whole team and the organization. Efficient, motivating Product Owner will get more results, correct results, and faster results. Good product ownership ensures that the team continuously works on the most important backlog items, gets them refined and specified optimally, gets immediate feedback, and delivers continuous high value stream.

Developing product ownership has impact

As important as delivering constant high value, is the development of the team capabilities and performance, and the team wellbeing. World Class Product Owner works together with Scrum Master and the team to ensure that team is not stressed and tired and can actively develop the tools, ways of working and the team assets. Product Owners have huge impact on how the team feels, not least affected by how Product Owner acknowledges good work and results, and takes the role of approval.

Training investment payback is fast

The cost of software R&D activities is high. A team of 10 people costs ~100kEur per month. Even an efficiency gain of 1% results in savings of over 10 000 euros annually. Developing Product Ownership to world class levels can result in efficiency gains of ten percent or more.

Expert trainers

Contribyte World Class Product Owner –training program is unique among Product Ownership training solutions. It is fully customizeable, training modules can be selected and expanded with company specific case studies, group work sessions and examples. The trainers have extensive experience in their expertise area.

Unique training program with immediate effect to daily work

The training solution consist of 9 modules of 3 hours each, delivered over a duration of 2-4 months. The 3 hour modules are easier to locate in the busy schedule of Product Owners than full day training courses, and they give focused information and skills that is instantly applicable to daily work.

Always customized, company situation specific delivery

World Class Product Owner –training program is always customized and delivered only to your organization.The challenges of each organization are different, and the people are different. Product Owners work with people. The best impact is achieved with a training program that contains company specific practical examples and case studies.

VALUE PROPOSITION

- Right features, right specification, right time, right release scope
- Projects progress visibility improved
- On time releases
- Less projects over schedule, budget and delayed
- Happy customers
- Better team spirit and higher state of self-organization in teams
- Less employee turnover
- Efficient communication to team, stakeholders and customers

Training solution details:

Online:

- 9 x 3 hour online training sessions
- max 12 participants

Classroom:

- 3 full day sessions or 6 half day sessions
- max 16 participants

Ask for pricing – contact

sales@contribyte.fi

ADVERTISEMENT: YOU WANT THIS, BADLY. EMAIL US. RIGHT NOW.

WORLD CLASS PRODUCT OWNER

COURSE OVERVIEW

Learn the best practices in different knowledge areas of experienced Product Owners. The World Class Product Owner -training is a practical, modular training which offers knowledge and tools for Product Owners with a few years of PO experience.

Available online sessions
(each module 2,5-3 hours)

Main modules

1. Introduction: How to become World Class PO
2. User Needs and Prioritization
3. Drive for Customer Value
4. Backlog Management
5. Epic and User Story Writing & Splitting
6. Agile Super Ceremonies – Backlog Refinement & Sprint planning
7. Agile Super Ceremonies Demo, Sprint Review & Retrospective
8. Managing the Scope & Stakeholders and Working with Other Teams
9. Working with the Team

Optional modules

10. Leader as a Team Coach
11. Drive for Implementation Quality
12. Improve Delegation and Team Self-Organization

VALUE PROPOSITION

- Right features, right specification, right time, right release scope
- On time releases
- Better team spirit and higher state of self-organization in teams
- Efficient communication to team, stakeholders and customers

Contribyte

Summary – Backlog Cheat Sheet

Clarify, stand up and enforce roles and rules

Communicate BL actively

Focus and keep top of BL well refined. Scan the rest regularly

Motivate team to Do good backlog refinement

Keep BL size under 150 items

Organize larger backlog With EPICs and themes

Practice story mapping

Set up dashboards

Screen bugs daily

Further Reading

Useful links for a new Product Owner:

- Agile Ceremony Improvement Guide **EN**. Link to the material.
- 9 tips for great backlog grooming **FIN**. Link to the webpage.
- Why is it important to learn how to split backlog items **FIN**. Link to the webpage.
- Preparation work makes backlog grooming more effective **FIN**. Link to the webpage.
- Contribyte's World Class Product Owner -training. Link to the webpage.

Books for a Product Owner:

- Marty Cagan: Inspired: How to create tech products that customers love
- Daniel Coyle: The Culture Code: The Secrets of Highly Successful Groups
- Simon Sinek: Start with Why
- Shawn Achor: The Happiness Advantage
- Arto Kiiskinen: 8 simple secrets of product owner success

About Contribyte

Contribyte is the coach of winning product organizations and teams of the future. We help organizations become better in product development.

Contribyte also trains product leaders, product managers, product owners, scrum masters and development teams. Our coaches have years of experience in improving product development team practices and product creation processes.

Read our blogs and subscribe our newsletter here: https://contribyte.fi/ajatuksia/tuteblogi/

Contact information
Feel free to contact us, if you have any feedback, comments or questions:
arto.kiiskinen@contribyte.fi
Info@contribyte.fi

Document Author:

Arto Kiiskinen
Senior consultant

Agile Coach Team of Contribyte:

Henri Hämäläinen
CEO, Principal consultant

Timo Leppä
Principal consultant

Camilla Magnusson
PhD, Senior consultant

Mikko Korkala
PhD, Senior consultant

Risto Varetsalo
Senior consultant

Contribyte

www.contribyte.fi