The Professional Product Owner

Don McGreal and Ralph Jocham
Chapter 8, part II

Release Management

Tomasz Kuks
Manager Technical Product Management

22 December 2021

01 Budgeting

02 Governance and Compliance

03 Kickoff

04 Quality

Source some of images: **“The Professional Product Owner”** by Don McGreal, Ralph Jocham

Budgeting

Be wise enough not to be reckless, but brave enough to take great risks.

—Frank Warren

PM – phase driven ORG

- Project Managers follow stages
- Prepare the budget, approve the budget, execute (and control) the budget, evaluate the budget
- Problem: money and date are set before actual work, when team knows the least of what to build

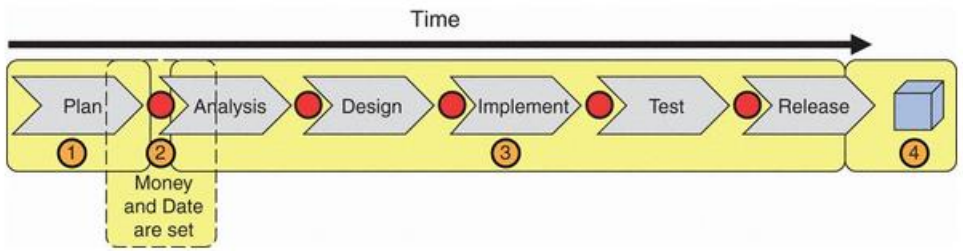


Figure 8-23 Setting date, time, and budget when the least knowledge is available

Agile budgeting – empirical evidence

- Leveraging emergence by collecting real data by building a small part of product
- This PoC should address anticipated features or technical risks
- The cost of learning predictable and often is less than time wasted on a plan and budget
- If risks are too high, then project is stopped, some $ is lost but it is controllable

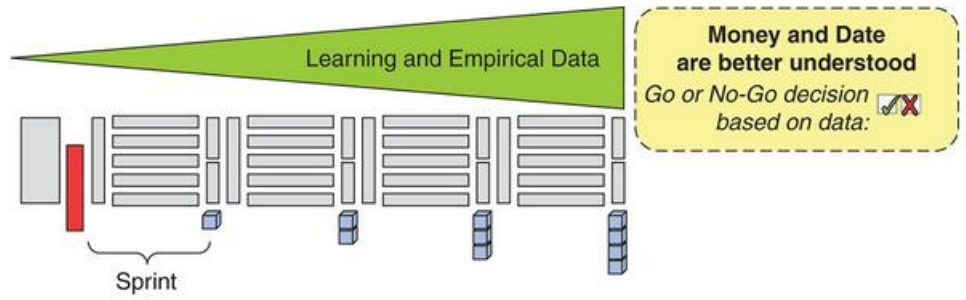


Figure 8-24 Two-phased budgeting: commitment after real knowledge is acquired

Real life case

I was on a Scrum Team for an internal call center application.

Those of us on the Development Team had no idea that our budget was almost up. However, a few Sprints earlier, when we were demonstrating some new functionality, one of the users convinced us to give them the functionality sooner, saying it would save him time during his daily tasks.

When it was later announced that our budget was up, that user was able to show how much time (money) the team saved with the new functionality, which made acquiring new budgets much easier.

Five steps for agile budgeting (FEED-ME)

- Fund product and their visions instead of projects

Knowing generated revenue and cost of running the product can lead to calculation of how many Dev teams can be funded

- Empower the Product Owner

Instead of assigning scope, schedule and budget to a PM, allow PO to be sponsor and have responsibility

- Establish transparency

Establish feedback loops, keep asking *"Are we still on right track?"*

- Demonstrate value sooner

The more often releases are set, the more often stakeholders will see a return on their investment, making them much more eager to continue or increase funding

- Manage stakeholder expectations

Ensure that stakeholders are consistently informed on development plan and progress. Be on the lookout for new a-h as the work progresses

- Employ empirical budgeting through validation

Instead of being driven by fixed budget (and scope and schedule), acknowledge that budgets may need to be reassigned, decreased, increased or killed. Plan to revisit budget often

When ORG is demanding upfront budget…

… and you have some say in the budgeting process:

- Build a Product Backlog
- Determine a potential velocity of Dev Team
- Determine number of sprints
- Multiply the number of sprints by its cost
- Realize that as you release functionality that provides a return, getting more funding should be easier

… and you have no say in the budget

- #1, #2
- Determine how far down the backlog you can get with potential velocity
- #5

… and you are asked for a fixed budget for a fixed scope

- Realize that you have been asked to take on all the risk, therefore communicate that the cost will be higher to effectively manage the risk
- #1, #2, #3,
- Realize that as you release functionality, customer will want changes. Accept any changes if there is capacity or if they can be swapped with equally sized items
- Ensure that the budget incudes a maintenance period following the end of initiative as there will always be unforeseen work after a release

Conclusion:

Fixed budget limits agility

Real life case

- (1) time and material costs, which allows the client to increase or decrease budget as they see value along the way;
- (2) a higher fixed cost, when they shift the risk to us.

Ultimately, the best thing you can do to manage all this risk is to build a working “Done” product at the end of every Sprint.

By acting in a way to produce value each sprint, the fundamental discussions around budgets and timelines change from **“Are we going to make it?”** to **“Are we getting the best ROI out of each sprint?”**

Governance and compliance, good way to look at it

The first two values of the 2001 Agile Manifesto are:

Individuals and interactions over ***processes and tools***

Working software over ***comprehensive documentation***

- Agile teams mitigate risk by creating working software each sprint, therefore documents, sign-offs and audits are often seen as redundant and that is an oversight

Table 8-4 Comparing Common Documents in Product Development

Requirements	User Guides
Business Rules	Training Materials
Test Cases	Legal Compliance (Sarbanes-Oxley)
UI Mock-ups	Support and Maintenance Guides
Designs	Security Compliance
Coding Style Guides	Legal Traceability-Matrix (FDA, FAA, etc.)

- Docs on left are for Dev Team (as items in backlog or in DoD)
- Docs on right are for s-h (as items in backlog or in DoD)
- Dev Team should select docs they need
- PO should select documents helpful for stakeholders

There are two main reasons for internal governance

Lack of trust with the Development Teams

…so they are asked to document their work to ensure they do not get off track

It is addressed with Scrum by creating working increments each sprint

Organization may want consistency between teams, products and departments

… to ultimately lower the costs

The larger the enterprise, the more governance is necessary to maintain oversight and stay in control

Agile governance

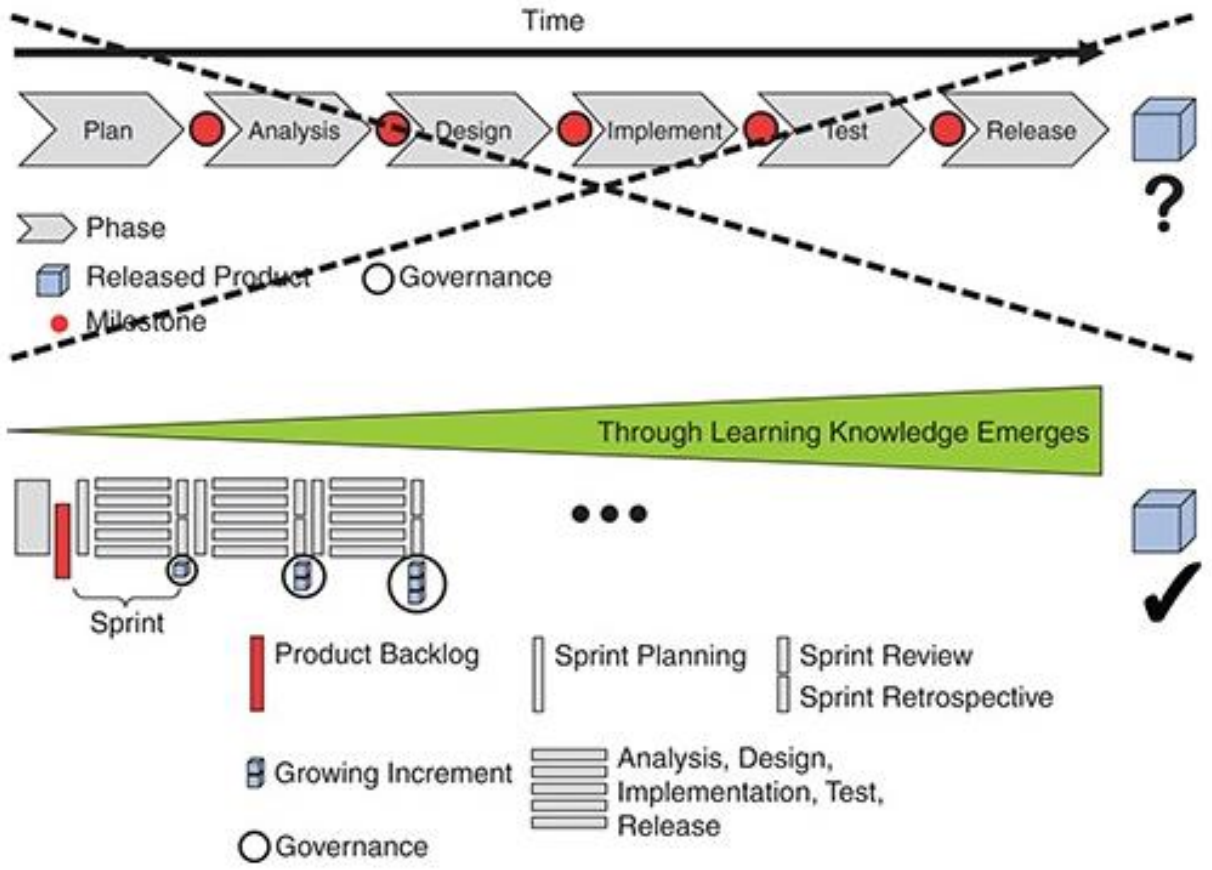


Figure 8-27 Agile governance based on working product

Agile A4 sprint report

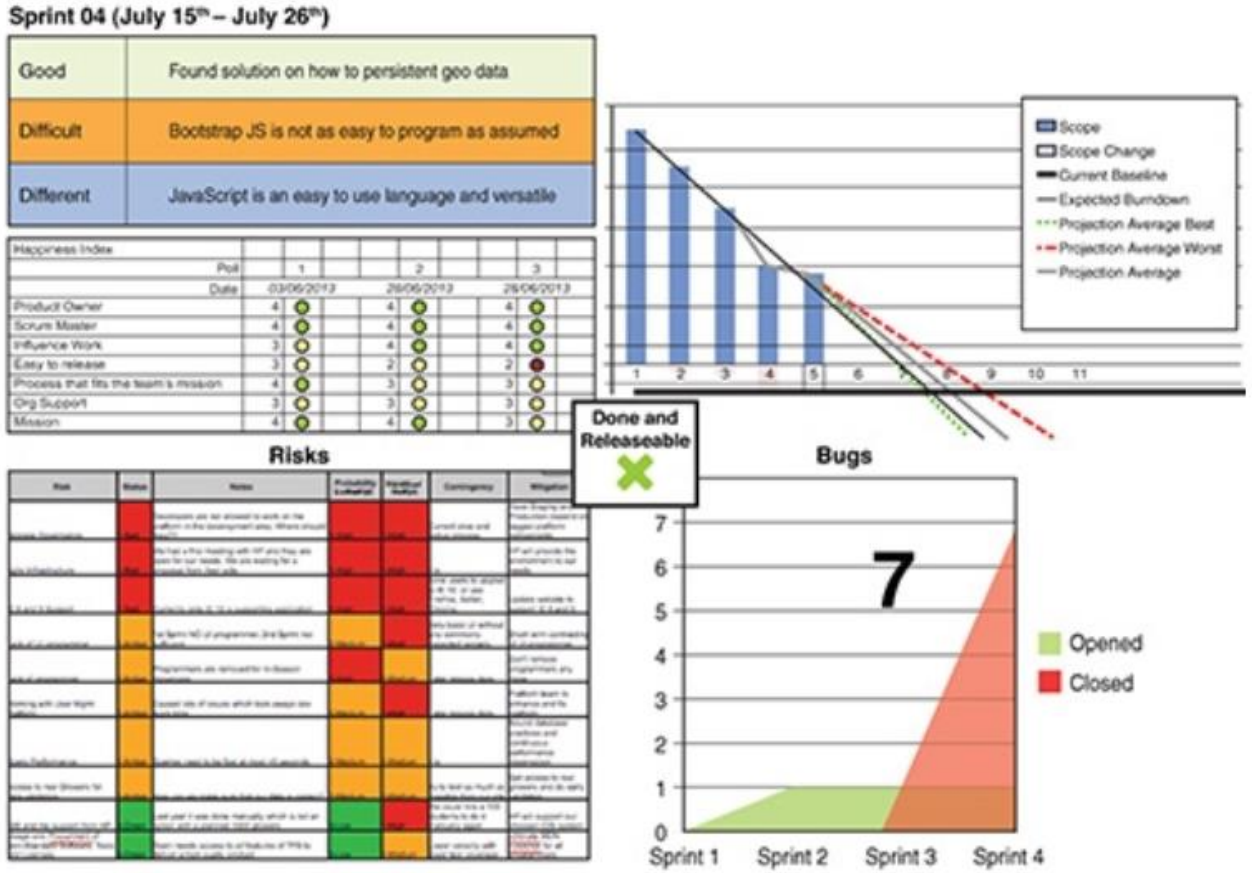
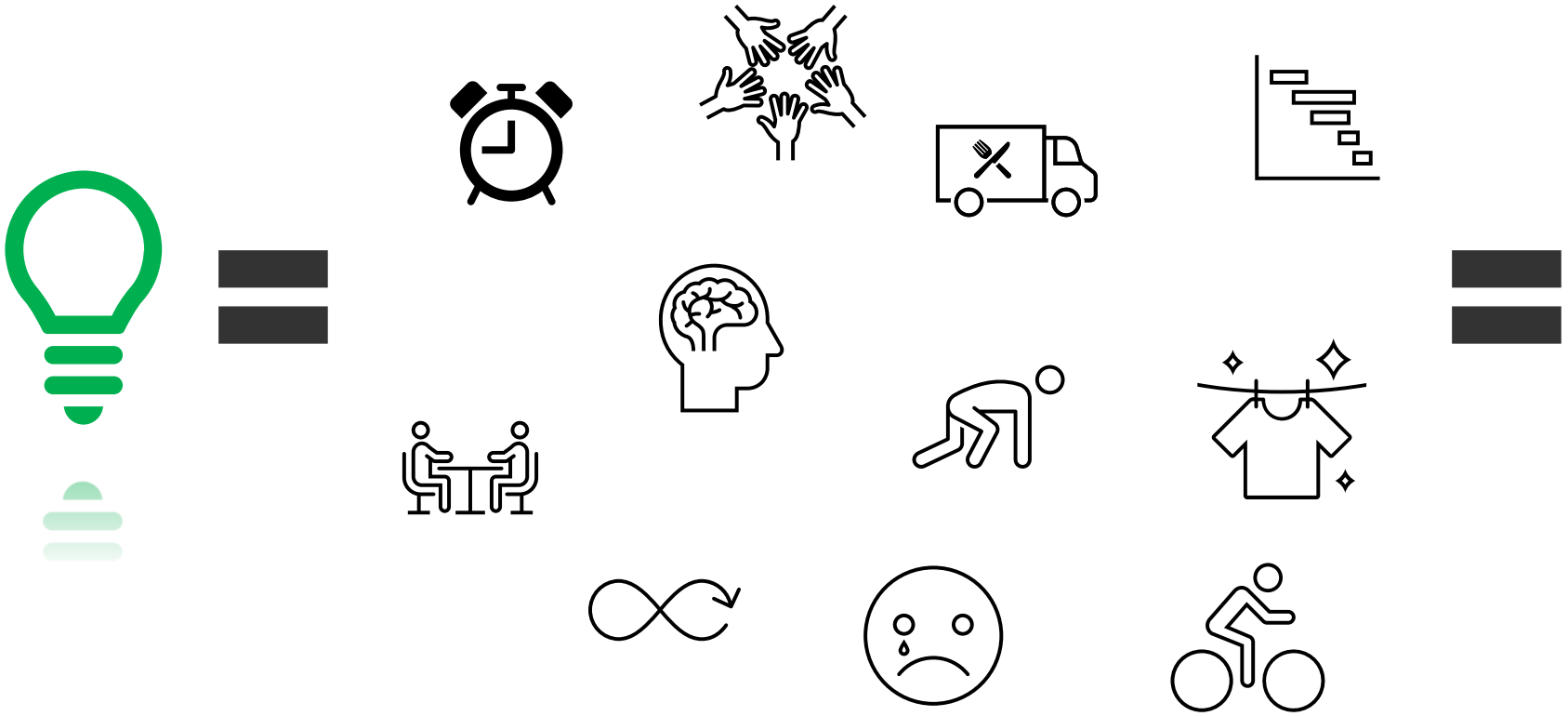


Figure 8-28 Agile A4 Sprint Report

Kickoff, …when planning a release

30% of success depends on how the team is launched



Kickoff with…

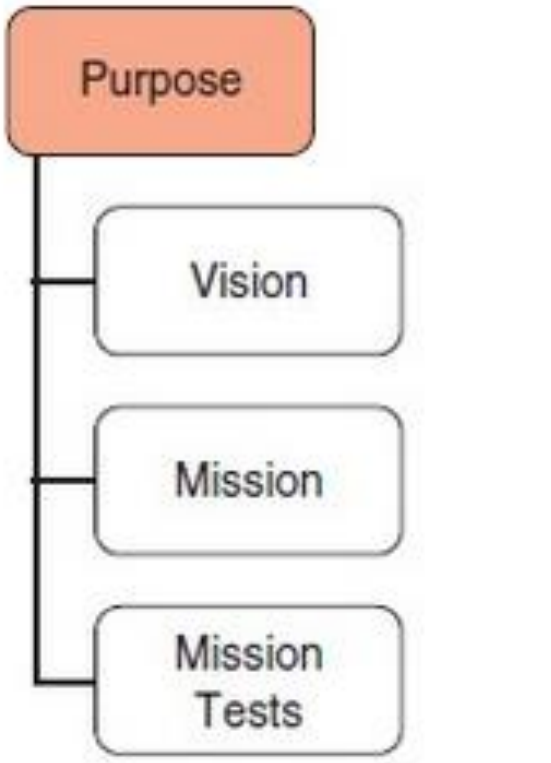
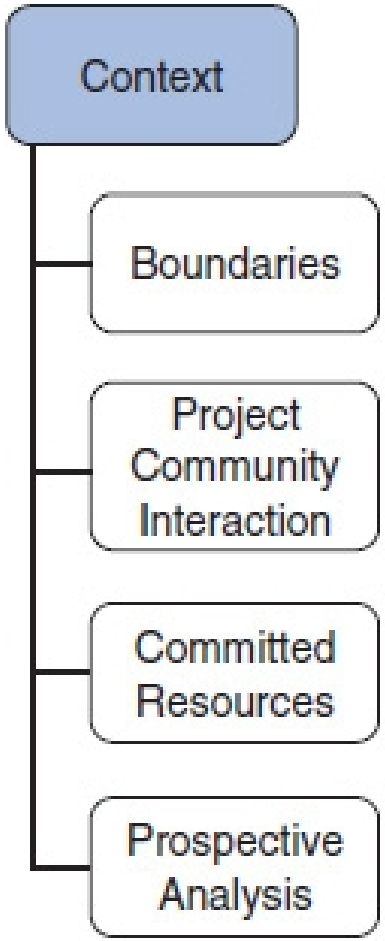
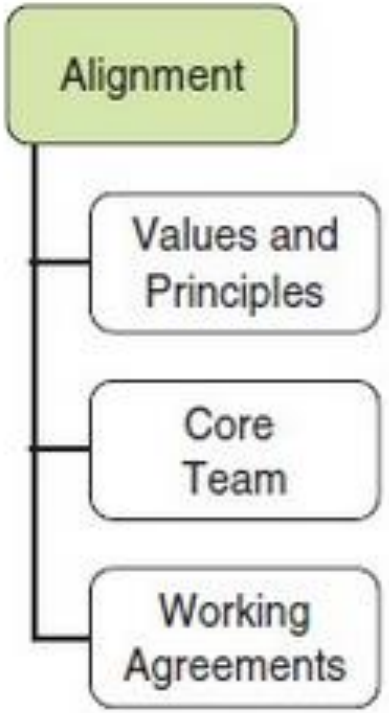


Figure 8-30 Purpose





Quality

…it focuses on process conformance and not on value and end user happiness.

… (types of quality) - product quality (PO - validation) and technical quality (Dev Team - verification)

Figure 8-34 Iron triangle of quality and relation to quality with the Scrum roles

Table 8-5 Possible Definition of Successful Project

#	On Time	On Budget	All Features	Status	Happy Customer
RT1	Yes	Yes	Yes	Success	?
RT2	Over	Over	Less	Challenged	?
RT3	?	?	?	Cancelled	? (No)
4	Ok	Ok	Less	?	Yes

RT1 – project success = on time and on budget

RT2 – project challenged = completed and operational, over budget, over time, fewer features

RT3 – project impacted = cancelled after some development

4 – Agile = most important value is discovered along development

Boosting business quality

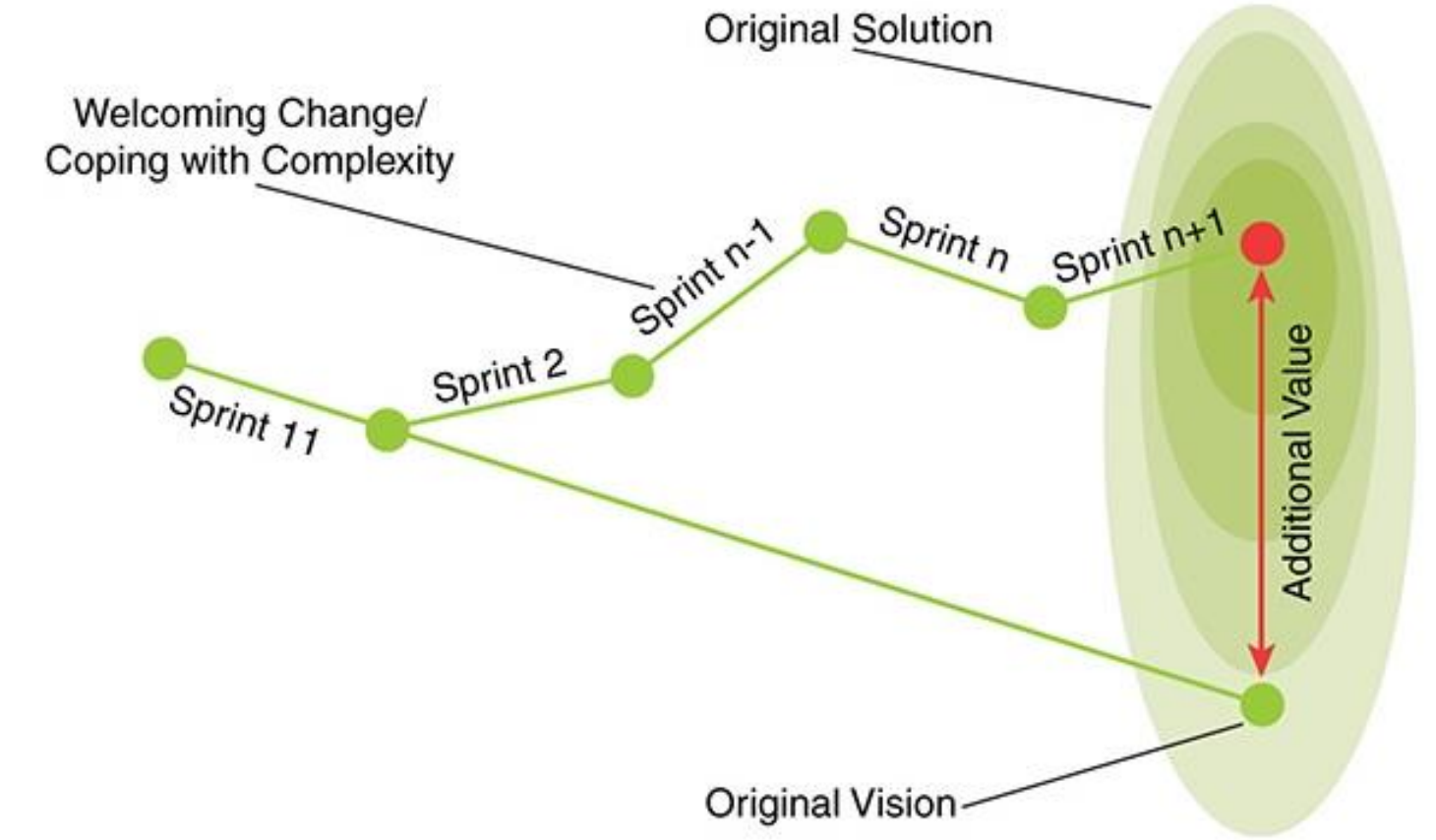


Figure 8-36 Maximizing and keeping value by ensuring quality

Keeping technical and business quality

Quality needs to be built into the product, from day one

Testing after the fact is about stability and not about quality

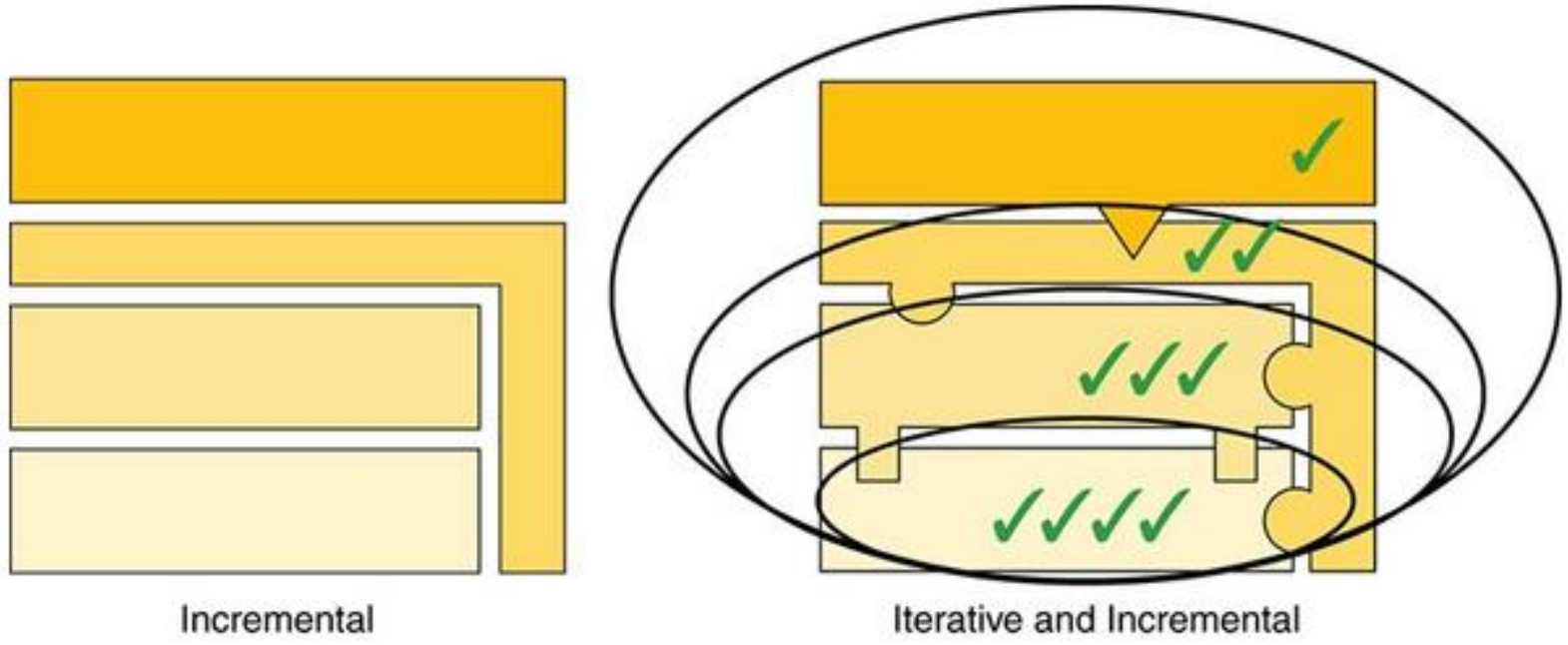


Figure 8-35 Keeping quality over time

Key Points

Budget with ROI

- **Clean up product backlog select features with potential**, coordinate with your Product Manager and focus on value
- **Use learning and empirical data** to try out solutions (PoC) to get budgets

Chaos control

- **Review document types you need to have/create/update in your product** for efficient product development and maintenance
- Each Sprint Planning make sure that team knows what value is going to be built

Key Points

Kickoff

- **Prepare and/or be prepared to lead kickoff events**, it should include **purpose, alignment, context**
- **Use learning and empirical data** to try out solutions (PoC) to get budgets

Quality

- **Mind product quality** – right set of features and functionality
- **Support technical quality** – limit tech debt, adequately tested, ready for release