

Product Integration Model Guide
For Agile Model

PIM-Go Agile

Version 2.1

Contents

THIS GUIDE'S OBJECTIVE	3
WHAT IS THE PRODUCT INTEGRATION MODEL (PIM-GO)?	3
Value-centered	3
Product-centered	4
Agile and Hybrid	5
PIM-GO MANIFESTO OF VALUES	6
Simplicity	6
Fail Fast	6
Breadth	6
Ease of Use	7
CULTURAL AND ORGANIZATIONAL ASPECTS	7
TOOLING ASPECTS	8
THE AGILE MODEL	8
PIM-GO AGILE'S ACTORS	9
Product Owner	9
Program Leader	9
Technical Specialist	10
Portfolio Manager	10
Scrum Master	11
Developers	11
Development Team Member (Conception Stage)	11
Scrum Team	11
Stakeholders	12
PIM-GO AGILE'S STAGES	12
Conception	12
Maturing	14
Sprints	15
Program	16
Portfolio	17
Discontinuation	18
ACKNOWLEDGEMENTS AND REVIEW	19
MAIN CHANGES IN THE CURRENT VERSION	19

This Guide's Objective

This guide was designed to support the understanding and use of the Product Integration Model (PIM-Go[®]) for the development of products and projects in agile environments.

Organizations that wish to promote, or are involved in, digital transformation processes will benefit from the contents of this guide. Fully or partially distributed teams will be able to use the model's processes and canvases to improve their results.

The model's simplicity and ease of understanding are meant to engage professionals at various levels and in various areas; it is not limited to the Information Technology industry.

This guide was created based on the model and information available on the PIM-Go website (www.pim-go.com), but is not restricted to them. The information on the website complements and provides details on the contents of this document, and both should be used together. Specifically for PIM-Go Agile, reading and understanding [Scrum Guide](#) is recommended.

What is the Product Integration Model (PIM-Go)?

The **Product Integration Model (PIM-Go)** is a set of models and artifacts for the development of products based on cutting-edge practices in product and project management.

PIM-Go is currently comprised of **PIM-Go Agile**, for products in highly changing environments, and **PIM-Go Hybrid**, for more predictive contexts, as shown in the figure below.



Figure 1: Context of use of PIM-Go's models according to the expected degree of changes.

While there is great similarity between PIM-Go Agile and Hybrid, especially regarding values, cultural aspects and the Conception, Program, Portfolio and Discontinuation stages, **this guide deals with PIM-Go Agile only.**

The **use of PIM-Go is free, and it is publicly available to be applied in organizations and training institutions.** However, citation is required as its content is protected by international copyright law, which includes its models, canvases and texts.

PIM-Go is:

Value-centered

PIM-Go is a model that is centered on value delivery.

By value delivery we mean delivery of a specific benefit, less its associated costs. The potential value to be delivered is defined based on a previous perception by the organization itself, and it is called **value hypothesis**. Based on the product's delivery, the validation of the value created may or may not confirm the value hypothesis.

Both the delivered value and the associated costs can be measurable or immeasurable.

For example, when an apartment is delivered, there is delivery of value to its user (owner) in a measurable way (savings on rent the owner will no longer pay), but also in an immeasurable way (satisfaction in living in that neighborhood or in a larger home). The associated costs for the owner will be the costs of moving (only once), in addition to property taxes and condominium and utilities fees (recurring) if higher than previously incurred.

Whenever possible, efforts should be made to translate immeasurable value into measurable value, through indicators. In the example above, the degree of satisfaction could be captured through a survey with the target audience.

Therefore, value creation occurs whenever a specific benefit is delivered to the user, and also by reducing the associated costs for using that delivery. In the example above, lower condominium fees of the new apartment can be a value hypothesis to be confirmed with the new owners.

Two types of value can result from a delivery:

1. **Internal Value:** the value of the delivery from the organization's perspective. It is usually related to the organization's strategic goals, such as: increased profitability, revenue, market value, reputation or image; or, conversely, a reduction in risk, operating costs, among others;
2. **External Value:** the value of the delivery from the perspective of the user (who uses the product) or the client (who pays for the product). It is usually related to solving a problem or meeting a need of this audience. In the example above, it refers to the benefit the owner receives through the delivery of the apartment, less its associated usage costs.

Note that even if a product is created to serve an intra-organizational audience, there will be internal and external value to be created. For example, if an organization is developing a new internal HR system in order to reduce operating costs, the internal value will focus on aspects related to financial gains (reduction in operating costs). The external value will be related to the satisfaction of the HR system's users and other workers who interact with it.

As a general rule, the teams involved in the product's construction level emphasize the concern with the value created for users and customers (external value). At the program and portfolio levels, internal and external values are balanced in line with the organization's goals.

Product-centered

Although it works with products and projects, PIM-Go is a product-focused model.

By product we mean a solution (whether physical or digital) that will be delivered to a user for use. Unlike projects, which are temporary, products have a more permanent temporal nature, although they may be discontinued at any time.

Even though it is also used for projects, we recommend adopting a more product-centric mindset in order to encourage a value-delivery culture.

This product-centric mindset refers to a departure from the predominant concern with delivery within scope, deadline and cost, towards a vision of value. In this context, a successful

product is one that actually solves a user's need or problem, even if changes occur during its development.

Note that the product-centric mindset should not be understood as using resources indefinitely. Products in the final stage of value delivery, i.e., with the user's main needs already solved, will only be allocated the resources necessary for maintaining the value that has been created or for taking advantage of some opportunity to create value that has not yet been explored.

Agile and Hybrid

In an organization, complex products and projects can have a wide range of contexts.

On the one hand, some initiatives have a high degree of stability of requirements and security regarding the solution's construction, and no major changes are expected during development. In these environments, establishing a thorough plan from project's outset is generally very useful for a successful execution. This context is called predictive and is usually associated with waterfall project development models.

On the other hand, there are initiatives with a high degree of uncertainty regarding the requirements or the construction to be carried out. In many situations, there may be no consensus on what should be built. And even if there is a set of initial requirements, there is also a high probability that they will change, as if they were a moving target. Or, even if there is certainty about what should be built, there is no clarity on how to do it, and changes are expected throughout the process, due to the team's learning. In these environments, thorough planning at the outset is not useful, since frequent changes will quickly make the plan obsolete. In these contexts, which are the opposite of predictive, an agile approach is usually recommended. **PIM-Go Agile** is a model aimed at this environment. To this end, the model uses consolidated agile practices, especially Scrum, and combines them with conception and management techniques for programs and portfolios.

PIM-Go Hybrid is a model aimed at building products and projects that have **some degree** of uncertainty. In this model, while there is consensus on the initiative's goals, some changes are expected, whether in the requirements or in the team's learning regarding how to build the solution, though to a limited extent. Thus, initial planning is still useful here, though without the thoroughness inherent in the waterfall model. Indeed, the degree of planning will be just enough for developing the first cycles, according to a similar concept called "rolling wave". On the other hand, the hybrid model has events and artifacts from the agile context which enable constant inspection, adaptation and value delivery, with reduced costs associated with change management. Thus, the essential practices of agile development are combined with those of the predictive context, becoming hybrid.

PIM-Go Manifesto of Values



Figure 2: PIM-Go values.

PIM-Go is founded on 4 values:

1. Simplicity;
2. Fail Fast;
3. Breadth;
4. Ease of use.

Simplicity

“We believe in practices that are simple, understandable for those who lead or participate in them.”

There are several practices to be used in product development. Without a clear understanding of those, there will be a lack of team engagement and less value delivery.

Thus, PIM-Go seeks to be simple by filtering what is essential for developing an outstanding product through an understandable flow (Big Picture). Particularities of each organization can and should be incorporated as needed, though without losing sight of the final simplicity. Complex workflows for dealing with exceptions should be avoided and direct communication promoted.

Fail Fast

“We believe that failing is inherent in the learning process, but it must be anticipated as much as possible.”

“Fail Fast” is directly related to “Learn Fast”. Organizations adopting PIM-Go should create environments that foster experimentation, validation and learning, preferably as early as possible.

Thus, PIM-Go includes stages and activities specifically designed for this purpose. As an example, the Conception stage uses ideation, design and feedback techniques to validate value hypotheses at an early stage, when few resources have been spent.

Breadth

“We believe that products have different market characteristics and contexts and should therefore have different approaches.”

PIM-Go is not a methodology, but models and techniques that can and should be adapted by organizations according to the characteristics of products and projects to be developed. Both models, PIM-Go Agile and PIM-Go Hybrid, cover largely recurrent situations in product and project management.

However, the nature of a product or project does not necessarily imply its adoption in all the organization's initiatives. As an example, PIM-Go's program and portfolio management is comprehensive to incorporate different initiatives developed with predictive, hybrid or agile processes.

Ease of Use

"We believe that theoretical models should be associated with guidance and artifacts that facilitate adopting them quickly in practice."

The lack of practical guidance on processes is also one of the reasons for decrease in the value created by teams. Thus, PIM-Go has contents integrated in the model that detail each stage in order to facilitate its use in organizations.

In many cases, these stages are supported by templates and Canvases with step-by-step instructions. This does not imply, however, that other techniques or adaptations cannot be used. On the contrary, throughout PIM-Go's flow, teams and organizations are encouraged to choose the technique that relates most closely to their characteristics.

An example is Risk Analysis. While there are several techniques for this purpose, PIM-Go suggests adopting a model based on three Canvases (Assumptions and Constraints, SWOT and Analysis). However, if the organization is familiar with another technique that achieves the same goal, that technique can be used.

Cultural and Organizational Aspects

Cultural and organizational aspects can boost PIM-Go's implementation or, on the contrary, they can create barriers or impediments to it.

To a greater or lesser extent, attention to the following is recommended:

- **Innovation Environment:** the existence of an environment that fosters experimentation, accepts failure and encourages learning is essential for teams to accept testing and innovating without the perceived risk of punishment;
- **Open Communication:** people should be able to communicate without hierarchical silos. Direct communication should be encouraged and information of common interest radiated to everyone involved without intermediaries. Tools should be integrated whenever possible to facilitate consistent, instant communication;
- **User-centered Vision:** the organization's processes should aim to create value for the user and constantly collect feedback. The collected data should be routinely stored and transformed into information and provide the grounds for decision-making;
- **Team Empowerment:** teams should have autonomy and power to decide the best way to build product increments. Processes that influence teams, such as budget and rewards, should preferably be aimed at the team and not the individual. This will encourage teams to organize themselves more effectively and establish a culture of commitment to the desired results;
- **Inspiring Leadership:** encouragement to change should be routinely reinforced by managers at various levels, stimulating value creation and impediment removal.

Leadership should be strategy-oriented and focused on creating a sense of engagement and purpose needed by the teams.

Tooling Aspects

PIM-Go is suitable for implementation through digital tools (software), and all the collaborative activities in the model can be carried out both in person and remotely.

For example, the OKR activity included in the item “Strategic Objectives” can be carried out in face-to-face workshops or through visual collaboration tools (a.k.a. Whiteboard tools), or both. The agenda and dynamics provided for the activity should be adapted according to the chosen means.

The use of digital tools makes it easier to capture and move data between the model’s teams and levels in a quick, consistent and effortless manner. This helps the organization manage knowledge more efficiently. An example: using Digital Canvases allows recording the whole discussion and migrating it to databases and other systems automatically. This guide does not present, or direct to the use of, any particular tool.

The Agile Model

PIM-Go’s Agile Model is based on a map-like graph called Big Picture, which presents a step-by-step path for the development and delivery of products and projects.

The model assumes adaptations to be made by organizations according to their reality. While different techniques may be used, it is expected that each stage in the model will be, in some way, carried out, even if tacitly.

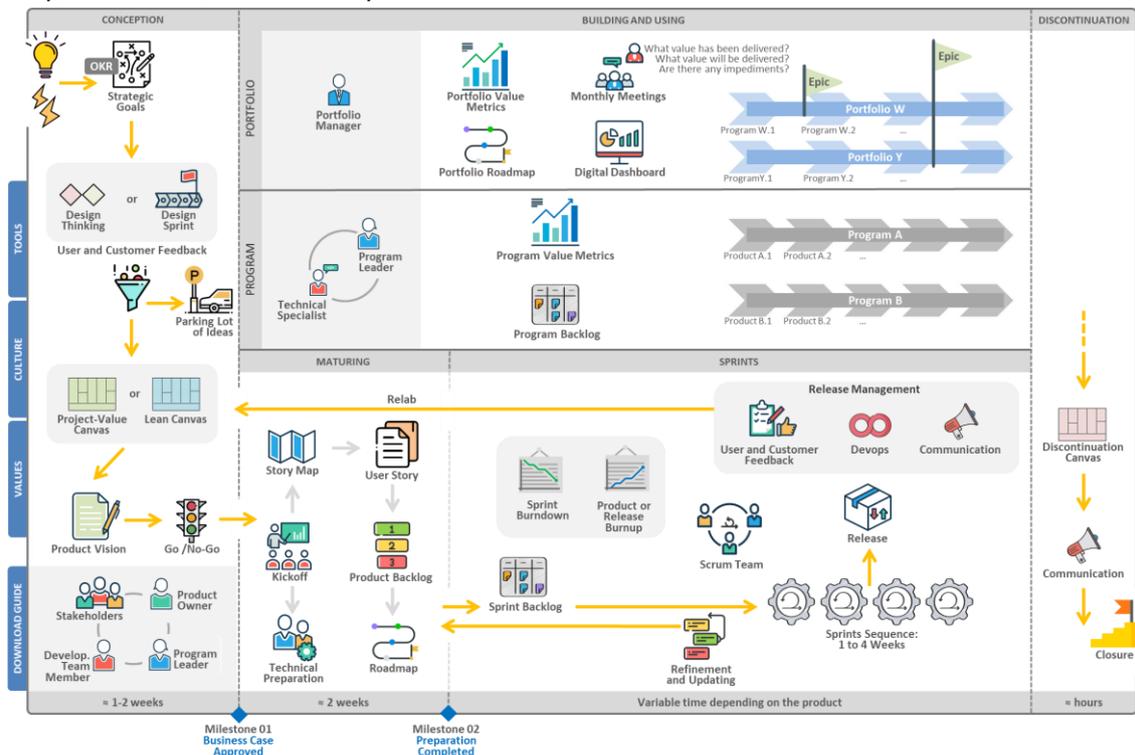


Figure 3: The Big Picture of the Agile Model.

PIM-Go Agile's Actors

Different actors participate in each stage in the solution's development.

Product Owner

Also known as P.O., the Product Owner is a role comprised in the [Scrum Guide](#). The P.O. is the person responsible for defining what product requirements should be implemented from the standpoint of value for the final user.

To that end, the P.O. must be constantly in contact with the product's target audience. In an agile team, the person in this role represents the user's and the customer's voice.

Ideally, the P.O. should know the business' domain, in addition to having some technical knowledge, in order to allow an effective communication with the Developers.

As also defined in the Scrum Guide, the P.O. should be one single person, not a committee. This prevents noise caused by inconsistent decisions about the product.

Duties

Among the Product Owner's main duties are:

- Maximizing the value of the product to be built;
- Keeping the [Product Backlog](#) updated, ordered and clearly communicated to the Developers.

The P.O. may choose to do the whole work directly or delegate it to others, though keeping the responsibility.

Program Leader

Often known as Product Lead, Senior Product Manager, Program Manager, or Head of Product, this position is common in organizations with multiple complex products and projects.

The Program Leader's main duty is to keep consistency regarding the program's multiple initiatives under a guideline that is aligned with the organization's strategic goals. To that end, the program leader can influence the priority given to specific product initiatives, if that impacts other products.

As indicated for the Product Owner in the [Scrum Guide](#), the program leader should be one single person, not a committee. This prevents noise caused by inconsistent decisions about the program.

Duties

The three main duties of a Program Leader are:

- Measuring and maximizing the program's overall (i.e. internal and external) value when considering multiple related products or projects;
- Keeping the [Program Backlog](#)'s consistency by keeping it organized, ordered and communicated to Product Owners and/or Initiative Leaders;

- Supporting the Portfolio Manager in identifying risks and opportunities of optimizing human and material resources within the Program.

Technical Specialist

The professional who works with technical aspects within the program. Their objective is to provide an integrated infrastructure across the different products to minimize risks and resources.

The name of this professional varies according to their industry, but it is always associated with a senior position. In the IT field, for example, the solution architect or enterprise architect usually takes on this role.

Duties

The main duties of a technical specialist are:

- Providing an integrated technical vision across the various products and projects;
- Creating new technical standards resulting from the initiatives' needs;
- Identifying technical risks related to the interdependence of initiatives;
- Proposing good practices related to technical aspects;
- Identifying resource optimization opportunities by reusing technical solutions across the multiple initiatives;
- Removing technical impediments that impact the program's delivery.

Portfolio Manager

The professional who manages integration across multiple programs and initiatives and their relationship with the organization's strategic goals.

The Portfolio Manager should be able to communicate to management and executive stakeholders the program's general status, the value delivered and yet to be delivered, as well as the associated risks. It is also the portfolio manager's duty to analyze the resources used and yet to be used, as well as to identify any opportunities for optimization or synergy.

Duties

The duties of a portfolio manager can vary from one organization to another but are commonly related to:

- Communicating and aligning the [Portfolio Roadmap](#) with senior stakeholders;
- Keeping coherence across the programs and checking their alignment with the strategy;
- Measuring the internal and external value promoted by ongoing programs and the expected future delivery;
- Promoting an integrated view of risks;
- Identifying and promoting any human and material resource optimization across programs.

Scrum Master

The Scrum Master is responsible for ensuring that Scrum is practiced within the team. Thus, the Scrum Master works so that Scrum events and concepts be applied in line with Scrum values.

The Scrum Master works through servant leadership, i.e., by serving the team so that it may produce the greatest possible value.

More details about the Scrum Master can be found in the [Scrum Guide](#).

Developers

Those responsible for developing the product. The Developers' definition is also detailed in the [Scrum Guide](#).

Important: The term *Developers* was created in the latest Scrum Guide replacing *Development Team*. As PIM-Go Agile is strongly connected to Scrum, *Developers* will be used predominantly throughout this guide. However, when *Development Team* is still mentioned (e.g. *Development Team Member* in the Conception phase), it always refer to the technical team responsible for product development. For PIM-Go Hybrid, the term Development Team is left unchanged as it is not directly based in Scrum.

Duties

The Developers are responsible for turning items included in the Sprint Backlog into increments potentially fit for delivery to users.

Development Team Member (Conception Stage)

In the Conception stage, this professional is a technical representative of the future development team to be formed if the product is approved in the "[Go/No-go](#)" stage. In addition to having a technical background, it is advisable for this member to be interested in the organization's business domain and processes.

This representative is necessary for two reasons:

- To incorporate a technical view in the discussion of the product, always associated with the value created by the product to solve a user problem or meet a user need;
- To facilitate the initial transition with the technical team in the Maturing stage. This role will be performed mainly during the initial weeks, when the team is not yet familiar with the solution and its context.

Scrum Team

It is the agile team formed by the Product Owner, Scrum Master and Development Team in Scrum. According to the [Scrum Guide](#)'s precepts, the Scrum team is:

- **Self-managing**, i.e., it has the autonomy to decide who does what, when, and how.
- **Cross-functional**, i.e., has all the required skills within the team to convert Backlog items into product increments.

The team normally has up to 10 members, including the Product Owner and the Scrum Master.

Stakeholders

Stakeholders are individuals who can somehow change or influence the potential value of a product or solution.

This influence can take place in the form of **value identification**. Product users and customers, UX team members, business unit managers directly related to the product are common examples of stakeholders who can assist the Initiative Leader in identifying higher-value solutions.

Another form of stakeholder influence is the ability to change the **delivery of value**. The project's financial manager, technical specialists required for delivering the solution, government representatives, or other members directly or indirectly impacted by the solution are also examples of stakeholders.

Duties

In the Conception stage, stakeholders participate as team members, usually with no hierarchical distinction. Their role, like that of any other member, is to contribute in the pursuit of the best solution to the user problem or need. Only in some very specific situations – such as in decision-making moments in Design Sprint – can the stakeholder, in the role of Decider, have a greater influence than that of other participants.

In the Building and Use stage, stakeholders are not formal members of the team and, therefore, do not have specific duties. However, they should be invited to key events, such as Review meetings and Roadmap presentations.

PIM-Go Agile's Stages

Conception

Conception	Portfolio		Discontinuation
	Program		
	Maturing	Sprints	

This is the stage of understanding the target audience, their problems or needs, and seeking solutions. Also in this stage, starting from a chosen idea, an initial detailing is carried out, which will be useful for obtaining authorization to proceed to the next phase (Maturing).

It is in the Conception stage that validation with users takes place so that inadequate solutions can be discarded and replaced quickly and with little impact. That is one of the practical implementations of the concept of Fail Fast, one of the PIM-Go values.

Why use it?

A solution that fails to solve the user's problem or need, even if well-constructed, is a bad solution. Therefore, the Conception stage is considered one of the most important in the PIM-Go model as it allows, through a range of techniques, discussing and understanding the best value creation alternatives.

How to use it?

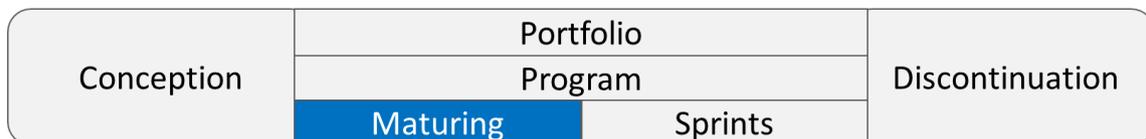
A working group should be formed which includes the Product Owner, the Program Leader, a senior member of the future technical team and the relevant stakeholders.

With the facilitation of an internal or external member of the group with experience in the process, the group should:

1. Start the process with a [need or problem](#), which may come from various sources. For example, a problem may be identified from a competitor's move or a need may arise from the customer service team. Problems or needs can also be identified based on the goals resulting from the [organization's strategic planning](#).
 - a. If the source is not the strategic goal, then the validation on whether the problem or need is aligned with the organization's strategy should be carried out at this point;
 - b. PIM-Go recommends using [OKR](#) (Objectives and Key Results) to identify strategic goals and related initiatives.
2. To identify a suitable solution to the problem or need, use an ideation method. PIM-GO's specific Design Thinking ([Design Thinking-Go](#)) or [Design Sprint](#) processes are recommended, but any other method can be adopted.
 - a. Whatever the method, interaction with the product's direct user is essential and should not be suppressed. Sometimes we tend to adopt preconceived solutions or adhere to certain value hypotheses that are part of our viewpoint. Only by directly consulting with users, through prototypes and user experience, can we confirm or dismiss those hypotheses, as well as expand our understanding of the solution;
 - b. *Design Thinking-Go* is an excellent option when there is some knowledge about who the target audience is. If that (i.e., the target audience, its pains or needs) is not clear, then previous stages for persona definition, empathy maps, user journey and other UX techniques are recommended;
 - c. If the option is a 5-day Design Sprint process, then problem definition is carried out on the 1st day (Define), ideation on the 2nd day (Sketch), selecting the winning idea on the 3rd day (Decide), prototyping on the 4th day (Prototype) and testing with users on the 5th day (Test);
 - d. If the option is Design Thinking-Go, then the detailing of the problem and its root causes should be done in the "Design Thinking - Causes" Canvas, ideation in the "Design Thinking - Ideas" Canvas, and selecting the idea in the "Idea Filter" Canvas. In the end, the selected idea should be prototyped and validated with users. Relevant unused ideas can be stored in the Parking Lot of Ideas ("Parking Lot of Ideas" Canvas).
3. Detail the selected solution using the "[Project-Value Canvas](#)", the "[Lean Canvas](#)" or another Canvas the organization prefers.
 - a. The "Lean Canvas" is recommended when the solution has the characteristics of a business unit, i.e., it competes for market share and has its own revenues and expenses. E.g.: startups;

- b. The “Project-Value Canvas” is recommended when the solution doesn’t need to compete for market share or generate its own revenue. E.g.: internal projects of the organization or in particular clients.
- 4. Consolidate information in the [Product Vision](#). This summarized vision is important for communicating with stakeholders and teams, and it is used throughout the building process as guidance on what is expected of the future product. The “Product Vision” Canvas is recommended for that purpose.
 - a. In Scrum 2020, this synthetic product description which provides a long-term goal for the team is called “Product Goal”;
 - b. As an alternative, some organizations may choose a more formal documentation model, such as a Project Charter. With no detriment to this approach, we recommend building the product vision also in this case, to guide the development carried out by the future teams.
- 5. Deliver a presentation of the solution to decision-making stakeholders based on the Product Vision and support documents (other Canvases). This stage is called “[Go/No-go](#)” and it is intended to filter initiatives at an early stage before more resources are mobilized (i.e., forming development teams, acquiring infrastructure, etc.). The expected result is authorization for the product to proceed to the Maturing stage.
 - a. This presentation is highly dependent on organizational culture in that it may be as simple as an e-mail, or sophisticated as a pitch for an initiative selection committee;
 - b. If approved, the initiative will formally enter the organization’s program and portfolio, with the respective associated value metrics.

Maturing



It is the stage of initial detailing of the solution within the “Building and Using” phase. The goal is to initially align the Scrum team and design the basic artifacts that provide an overview of what the solution will be, as well as its initial requirements.

By the end of the Maturing stage the teams should have the technical, human and material conditions to execute the first Sprint.

Why use it?

Without this stage, part of the first Sprint’s time would be dedicated to designing some fundamental artifacts, such as the Product Backlog, in a disorganized manner.

Using a previous stage for the initial detailing allows the team to discuss the solution at a higher level, starting with the Story Map from the perspective of the user’s journey, then the Product Backlog and, finally, detailing the user stories of the initial Sprints.

How to use it?

The Product Owner, with the Scrum Master’s help:

- Prepares and holds the [Kickoff](#) meeting with the Developers to align the work carried out in the Conception phase;
- Creates with the Developers the product's [Story Map](#), including the user's journey. The discussion will allow identifying Epics and stories to be included in the [Product Backlog](#);
- Defines the Backlog's initial order based on the Story Map, and, along with the Developers, conducts the Backlog's first refinement to detail the main [user stories](#);
- In the second week of Maturing, creates the [product's Roadmap](#) with the Scrum team to be presented to the stakeholders;
- Throughout the process, the Scrum Master and the Developers make [technical preparations](#) in tandem to make sure all resources and configurations are available in Sprint 1.

The steps above are not unidirectional, but bidirectional. For example, as the Roadmap is being defined, something relevant may be identified which was not included in the Story Map or in the Product Backlog. This way of working is essential in the agile model.

Sprints



It is the stage for converting Product Backlog items into increments, i.e., when the solution is actually built. The Sprint is a cyclical activity, i.e., at the end of one Sprint, the next one immediately begins. At the end of a Sprint, the Product Owner can choose to make a release to users.

Why use it?

Using Sprints within a predefined period allows quick course corrections, thus converging with the Scrum's inspection and adaptation pillars.

How to use it?

The sprint should be limited to a maximum 4-week period; however, 2- to 3-week Sprints are also common. Once you have defined what will be done in the Sprint (Product Backlog item selection), you shouldn't change the Sprint's content, except in extraordinary situations. At the end of the Sprint, a product increment should be generated which will be evaluated by the Product Owner with other stakeholders; the Developers and the Scrum Master should always participate in the evaluation. The feedback provided will feed the planning of the future Sprint.

The Sprint itself is a Scrum event. However, during its execution, other events are also carried out, namely:

- **Sprint Planning:** when the Scrum Team designs the plan that will be developed in the Sprint, translated into an initial [Sprint Backlog](#). It lasts up to 8 hours for 1-month Sprints;
- **Daily Scrum:** a daily meeting for aligning the Developers, limited to 15 minutes;

- **Sprint Review:** a meeting for inspecting the increment developed by the end of the Sprint and adapting the Product Backlog based on feedback from the P.O. and stakeholders. It is limited to 4 hours for 1-month Sprints;
- **Retrospective:** a continuous improvement meeting where the Scrum team reflects on its work during the Sprint. It is an opportunity for identifying what went well and what could be improved. It should last no more than 3 hours for 1-month Sprints.

Throughout the Sprint, the Developers should assess the development of its work and how close it is to the Sprint’s goal. [Burndown](#) graphs are recommended for that purpose.

In turn, the P.O. should assess the solution’s progress periodically, based on the increments generated in multiple Sprints. The [Burnup](#) graph is recommended for that purpose.

Further details on Scrum theory, roles, events, artifacts, rules and values can be found directly in the [Scrum Guide](#).

Program



The management level that operates with multiple interrelated products through a Program Leader and a Technical Specialist.

Programs are groups of initiatives that seek to deliver value which would not be available if those initiatives were to be independently conducted.

This analysis of created value may be externally oriented (regarding the user and client) or internally oriented (regarding the organization). At the program level, the goal is to maximize the value created by the initiatives together, as well as reduce their associated costs:

1. **External Value Increase:** an increase in the final value for the user or client by composing individual value deliveries or reducing their use costs. Example: applications related to the same target audience (e.g., accountants) and which may be used in combination (increased value resulting from the combination);
2. **Internal Value Increase:** an increase in the final value for the organization by composing individual value deliveries or reducing the organization’s costs. Example: initiatives related to sustainability which share the same management structure (decreased costs associated with managing the initiative and similar purchases).

Why use it?

For medium and large organizations, management just at the product level cannot provide a comprehensive view of the integration between multiple products. Using a Program-based approach allows taking advantage of synergies between different products, as well as aligning their actions to maximize the generated value.

In addition, analyzing at the Program level allows identifying risks in an integrated manner.

How to use it?

Coordinated by the Program Leader, Program management is constant and necessarily implies executing at least 3 activities on a routine basis:

1. **Periodical Program assessment meeting:** attended by the Program Leader, Technical Specialist and respective Initiative Leaders, it is aimed at aligning ongoing activities, identifying dependencies and program risks, as well as evaluating metrics and opportunities for synergy. It should be held at least fortnightly;
2. **Keeping a [Program Backlog](#):** i.e., creating and updating a Backlog that includes all initiatives in progress and a view of each epic’s status. It is usually in digital format and is sometimes integrated to the control systems of the development teams of related products;
3. **Keeping [Value Metrics](#):** i.e., creating and updating the program’s value metrics based on the metrics of each product. These metrics will also be provided for interaction by the Portfolio Manager at the Portfolio level.

Portfolio



The strategic management level that operates with multiple programs or key products through a Portfolio Manager.

Portfolios are groups of programs or complex products linked to one or more strategic goals.

Why use it?

For large organizations, portfolio management is the connection between strategy and ongoing initiatives, which allows alignment with the organization’s executive level.

How to use it?

Portfolio management is continuous, even if changes occur in the set of portfolios.

For an effective portfolio management, the Portfolio Manager should execute these activities on a routine basis:

1. **[Monthly meetings](#):** periodical meetings with the executive team to update the portfolio. These meetings are aimed at identifying the value that has been generated, the value to be generated, and related impediments. The mapping of changes in the organization’s external and internal environments should also be discussed so as to identify the necessary adaptations to the portfolio;
2. **Keeping [Value Metrics](#):** i.e., creating and updating the portfolio’s value metrics and directly linking them to the related strategic goals. The value metrics should become visually available to the organization through a [digital dashboard](#);
3. **Keeping a [Portfolio Roadmap](#):** this should provide a comprehensive high-level view of the programs’ activities and related initiatives, as well as their relationship with the associated value metrics. It is usually created in digital format to make it visible to

stakeholders regardless of their location; its language and format should be easily understandable.

Discontinuation



It is the stage conducted when a decision has been made to terminate the product or project. This stage is also applicable when the product is being developed for another company, and the agreement is terminated.

Why use it?

In product development models, as a rule, efforts are focused on the previous stages, which are viewed as value generators. However, terminating a product's life cycle can also affect the value perceived by users.

For example, when a product is being terminated, there may be an opportunity to offer the user an alternative as a way of minimizing the negative value generated by that termination.

In the case of projects, this stage occurs when the initiative is terminated, whether in a planned way (e.g., when all releases have been completed) or abruptly (e.g., through the sponsor's decision to terminate the project).

Thus, the goals of the Discontinuation stage include:

- Mapping who is potentially impacted by the Discontinuation and identifying the actions and opportunities to be taken for maximizing value generation;
- Defining the schedule of actions of discontinuation and the releasing of human and material resources;
- Promoting team reflection about the work done in the initiative, and collecting the lessons learned that can be used in other initiatives.

In many situations, strategic issues will lead to Discontinuation. It is important that the rationale for such decisions be written down in case the product is resumed, or just to clarify to the organization the trade-off enabled by the Discontinuation.

How to use it?

Once the decision to discontinue the product has been made, the Initiative Leader and the team and relevant stakeholders should:

1. Use the [Discontinuation Canvas](#) to discuss and list anyone potentially impacted by the discontinuation;
 - a. In addition to users and customers (as potentially impacted groups), employees and partners should also be included;
2. List on the Canvas any actions to maximize the product's value (or reduce any negative value) for those groups;
3. Capture on the Canvas the lessons learned from the initiative development journey which may be useful in future solutions;

4. Define and execute the necessary communication actions to keep the impacted groups informed of ongoing activities and possible alternatives.

The initiative's actual discontinuation only takes place once the actions related to discontinuation have been completed.

Acknowledgements and Review

This guide was designed by Professors Alvaro Junqueira and Igor Lagreca, but it is the result of the combined work of many professionals to whom we express our gratitude.

We are especially grateful to the reviewers who provided us with valuable considerations for our analysis. They are:

- i. Armando Oliveira, MSc., PMP[®], ASF[®], SFC[®], PSM[®] I, Prince[®] Foundation;
- ii. Enock Godoy de Souza, DSc., PMP[®], MoP[®], Cobit[®] 5 Foundation, Prince2[®] Practitioner, ITIL[®] Expert, PSM[®] I, CSPO[®];
- iii. Mauricio Arantes de Andrade, MSc., PMP[®], ITIL[®] Expert, AIC[®].

We also received considerations from the following professionals, who greatly contributed in the discussions on the management of projects, products, programs and portfolios related with the practices in this guide. They are:

- i. Jim Buchan, MSc (Hons);
- ii. Stefan Sohnchen, PMI-ACP[®], CSM[®].

Main Changes in the Current Version

In addition to other specific adjustments, the version 2.1 of the PIM-Go Agile Guide incorporated the main changes to the Scrum Guide 2020, especially the term Developers, which replaced the Development Team.