



e-book

Integration patterns: Architecting your value stream for speed and quality



“ How can you go faster if you don't know what's slowing you down? ”

Carmen DeArdo, author of *Standing on Shoulders: A Leader's Guide to Digital Transformation*

As traditional enterprises struggle to transform in the Age of Digital Disruption and compete with tech giants and startups, the need to accelerate the flow of business value through software is now a critical business priority. Leading organizations, many of whom are in the *Fortune* 100, have taken key steps to architect their software delivery value stream around flow to successfully reduce the time to value of their software products.

Architecting for flow is crucial because enterprise software delivery, by its very nature, is a broken process. The complex – and often implicit – network of teams, tools and processes that plan, build and deliver are disconnected by the role-specific tools used by teams across the **ideate, create, release** and **operate** stages. These excellent tools are built for purpose, such as planning, design, development, testing, service desk and so on, and enable specialists teams to do their jobs more efficiently and collaboratively.

Even for the many organizations implementing the Scaled Agile Framework (SAFe®) to scale their software delivery, the complex product work spans dozens of Agile development teams and a heterogeneous toolchain.

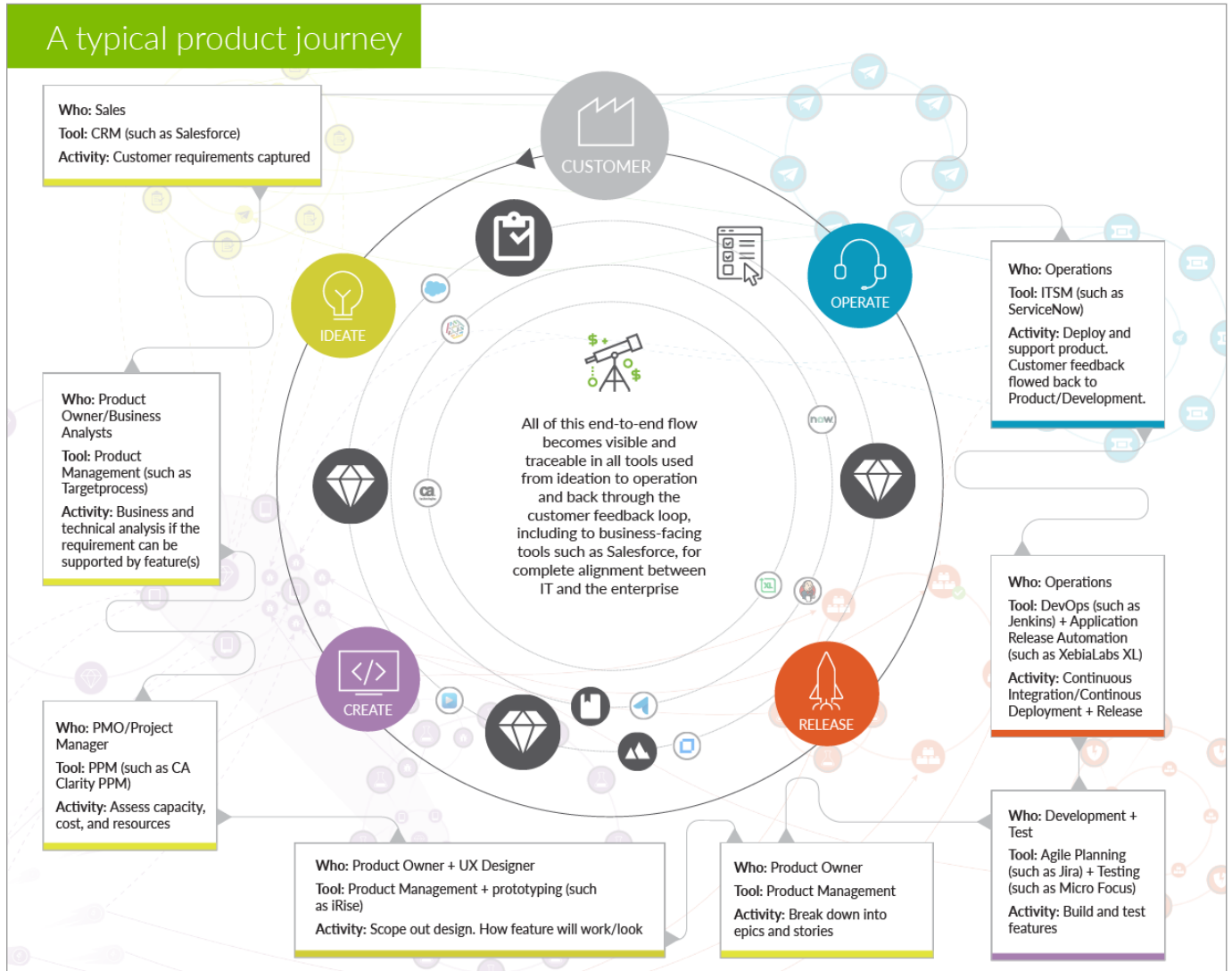
All these tools, however, are not designed to naturally work with other tools in the value stream, meaning they do not automatically flow the product-critical information that specialist teams need for collaboration throughout a product's development, delivery and support. Consequently, these time-strapped and under-pressure teams are forced into mind-numbing manual forms of knowledge sharing, such as:

- Email threads
- IMs
- Standing meetings
- Tool switching
- Duplicate entry
- Spreadsheet management

These archaic approaches are cumbersome, slow and susceptible to human error, and they provide no end-to-end visibility or traceability into the flow of work. Our own research has found that this **manual waste can steal up to 20 minutes per day, per specialist – equating to 10 working days per person, per year!**

With no formal process, it can be hard to see how work is flowing between stages and obtain one source of truth of a product's journey. If you can't see the workflow to identify bottlenecks, how can you gain control and take the steps to improve it?

By connecting this invisible network through a series of **integration patterns**, organizations can make the process explicit, automating and visualizing the flow of work from ideation to operation and back through the customer feedback loop. In doing so, you also enable all teams to work in their tool of choice, allowing a best-of-breed approach to flourish.



To create this automated end-to-end flow, many *Fortune* 100 companies are connecting their software delivery value streams by stitching common integration patterns together. In doing so, they're making the entire Value Stream Network visible, to reveal the whole story of the product's journey from a customer's lips to their screen. A story that traces all the work from requirement to delivery and back, helping product owners to better understand how the product was built, where the flow slowed down, what the end users think and how to swiftly improve any issues to deliver a more delightful experience.

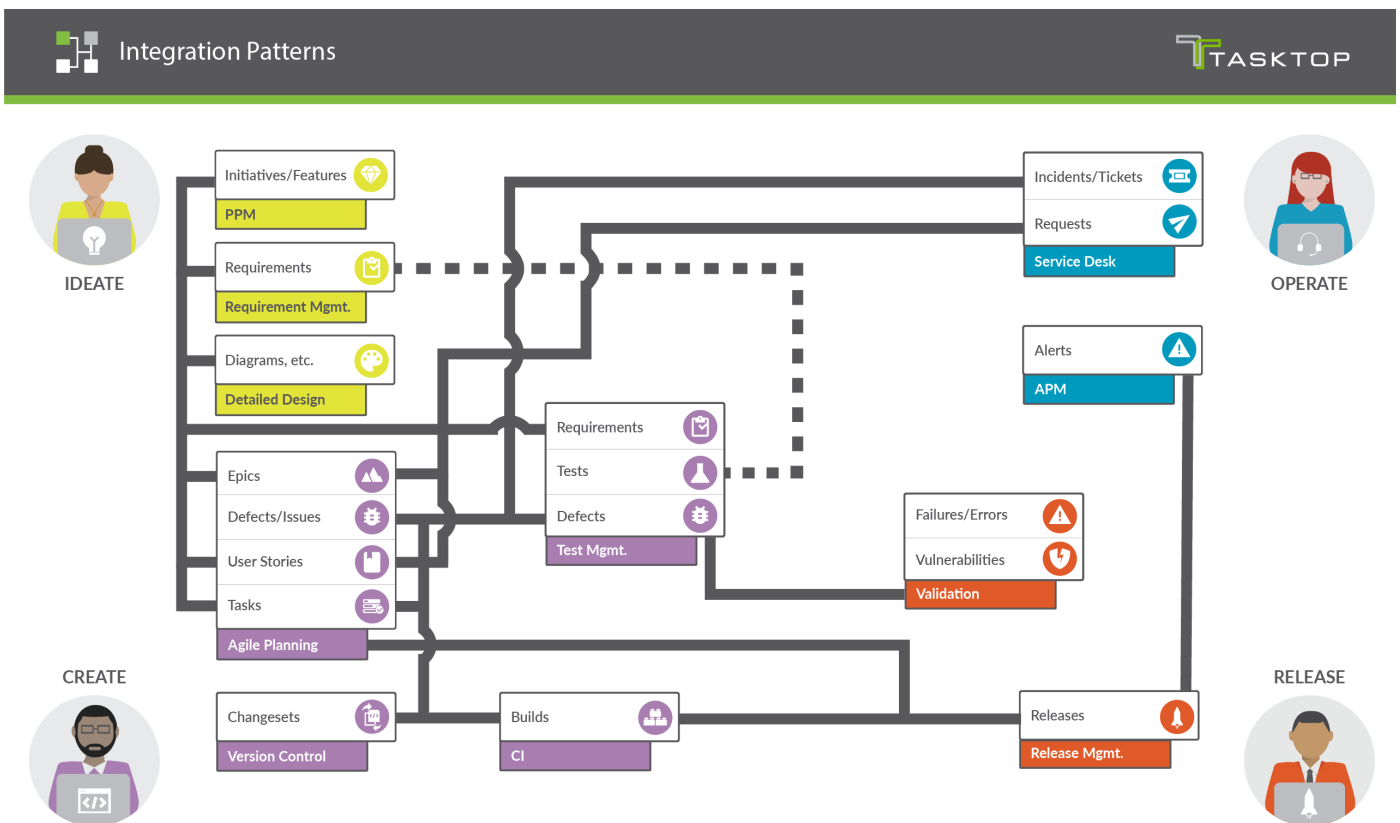
Integration patterns are **common** and **critical** interactions between the specialists and the tools they use to help plan, build and deliver software at scale. We have identified 14 so far - and that number continues to grow as enterprises find more sophisticated ways to eradicate waste, remove inefficiencies and accelerate flow between key stages of the value stream to meet specific business needs.

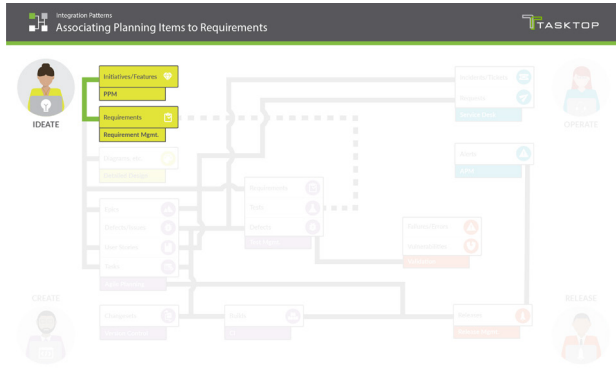
Benefits of integration patterns

- Eliminates overhead and duplicate data entry
- Accelerates delivery speed
- Enhances cross-team collaboration
- Improves time to value and mean time to resolution (MTTR)
- Enables end-to-end visibility and measurement of flow (including Flow Metrics)
- Creates traceability for compliance and continuous improvement

Common integration patterns

Here are **14 common integration patterns** that leading organizations are using. **The sophistication of these chained patterns has grown significantly since 2013.**





PATTERN 1

Name: Planning to requirement

Stage: Ideate

What: Brings the people who manage product workflow closer to the people who understand what the customer needs from the product.

What's flowing?

Planning Items	Requirement Items
Portfolio items	Epics
Initiatives	Stories
Features	Tasks

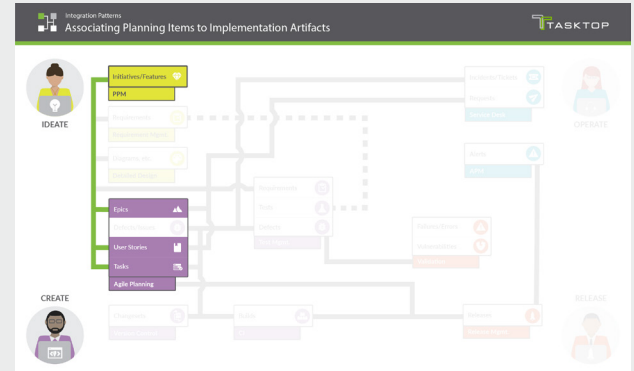
About

Some initiatives or features defined in PPM tools, or in product roadmapping tools, can represent requirements for the product and development teams. Organizations that develop very complex products in heavily-regulated industries (such as healthcare, finance, government, insurance) use Requirements Management tools for their product specifications.

This integration pattern mirrors the high-level planning items managed in the PPM or roadmapping tool directly into the Requirements Management tool, from which they can be broken down into requirements, user stories or epics.

Why

Creates a traceable flow of work for compliance.



PATTERN 2

Name: Planning to implementation

Stage: Ideate and Create

What: Brings the people who understand what the customer needs from the product, and the business value it will provide, closer to the people who are building it to ensure development is accurate, on budget and on schedule.

What's flowing?

Planning Items	Implementation Items
Portfolio items	Epics
Initiatives	Stories
Features	Tasks

About

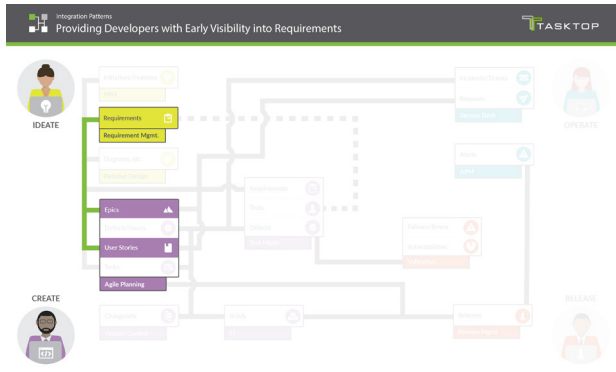
The Project Management Office (PMO) often uses a PPM tool to help manage organizational initiatives and investments in the application portfolio. In recent years, product-centric organizations have also adopted roadmapping tools for their strategic planning.

At the same time, the product and development teams tasked with executing against these high-level business themes manage their work in Agile planning tools.

This integration pattern mirrors the high-level planning items managed in the PPM or roadmapping tool directly into the Agile planning tool, from which they can be broken down into epics, user stories, tasks and sub-tasks.

Why

Work progression is flowed upstream to Product Portfolio Managers so the business is kept abreast of key software-centric driven business initiatives.



PATTERN 3

Name: Developer visibility into requirements

Stage: Ideate and Create

What: Provides the people who build the products with the business-critical information they need to accurately and rapidly build products that also adhere to stringent regulations.

What's flowing?

Requirement Items	Implementation Items
Requirements	Epics
User stories	Features
	Stories

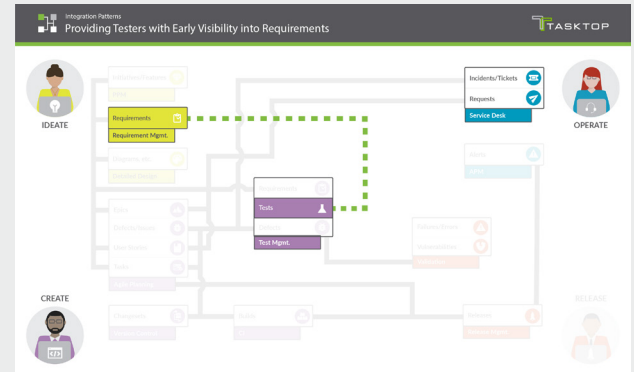
About

Organizations that develop very complex products in heavily-regulated industries use Requirements Management for their product requirements.

The development teams who need to implement those requirements would prefer to have them flow into their Agile planning tools automatically as epics or features. This information is then broken down into user stories, maintaining the relationship to the parent epic and its associated requirement for traceability.

Why

Data flowed back upstream to provide Portfolio Managers with visibility into execution and completion against business initiative.



PATTERN 4

Name: Tester visibility into requirements

Stage: Ideate and Create

What: Provides the people who test the products with the critical information they need to meet stringent regulations.

What's flowing?

Requirement Items	Testing Items
Portfolio items	Test cases
Initiatives	
Features	

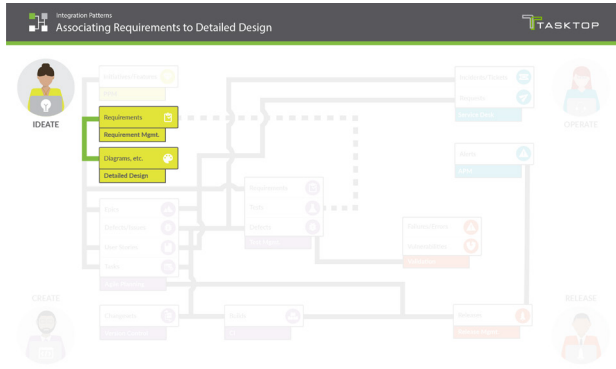
About

Testing and QA teams have a crucial role ensuring product quality and protecting brand integrity. Including them in the earliest phases of the software delivery process ensures they have ample time to work on the test cases that will cover the requirements.

This integration pattern mirrors and synchronizes requirements or user stories authored in Requirements Management tools to the testing tools, where test plans, test cases and test scripts are created. If tests are systematically linked to the originating requirement, you can easily identify which tests might need to be updated when a requirement is changed, or when functionality is removed from the product. It also provides the Product teams with visibility into test coverage as testing progresses, if supported.

Why

- Improves product quality
- Changes to a requirement are seamlessly communicated to QA teams
- Product Managers and Business Analysts have visibility into test coverage (completion) for each requirement
- Traceability is maintained between requirement and test for complete coverage and prevention of over-testing when specific functionality is removed



PATTERN 5

Name: Requirements to product design

Stage: Ideate

What: Provides the people who design the products with the critical information they need to meet stringent regulations.

What's flowing?

Requirement Items	Detailed design
Requirements	Requirements
User Stories	Use Cases
Epics	Actors
	Business Process Diagrams
	Documents
	Diagrams
	Storyboards
	UI Mockups

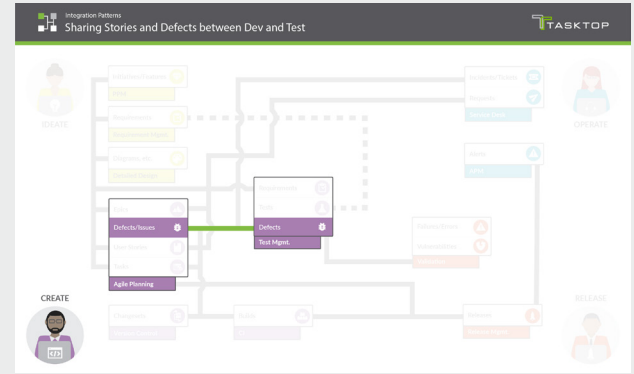
About

The process of defining and designing a product and its features often spans multiple tools. For example, requirements may be authored in a Requirements Management tool and then elaborated on in a modeling tool, where the customer experience and business processes are fully detailed and supplemented with wireframes, mockups, diagrams and more.

This integration pattern mirrors and synchronizes requirements from a Requirements Management tool into additional tools used for detailed design, like Storyteller or Sparx EA.

Why

- Product Managers, Business Analysts and Designers better collaborate to develop comprehensive and clear requirements
- The relationship between requirements and their detailed designs are fully traceable



PATTERN 6

Name: Development and test

Stage: Create

What: Brings the people who build the product closer to the people who test the product to reduce defects before and after product is in operation.

What's flowing?

Requirement Items	Detailed design
Requirements	Requirements
User Stories	Use Cases
Epics	Actors
	Business Process Diagrams
	Documents
	Diagrams
	Storyboards
	UI Mockups

About

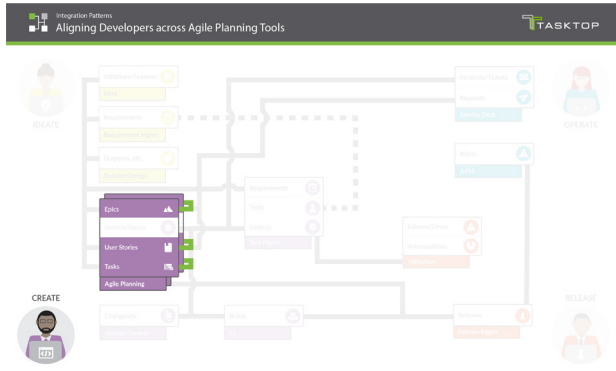
This is the most common integration pattern. By integrating testing tools with Agile planning tools, customers can focus on improving collaboration between development and test teams to improve product quality and mean time to resolution.

This integration pattern:

- Flows stories from Agile planning tools to testing tools, improving test coverage and traceability between stories and test cases.
- Flows defects from testing tools to Agile planning tools, accelerating time for developer to resolve issue(s). Once fixed, tester is immediately notified in their own tool to continue testing

Why

- Optimize testing efforts by providing early visibility into stories and tasks
- Improve product quality and speed of defect resolution



PATTERN 7

Name: Aligning Agile development

Stage: Create

What: Brings all the Agile Development teams together for cross-team visibility.

What's flowing?

Development Items
Epics
Features
Stories
Tasks

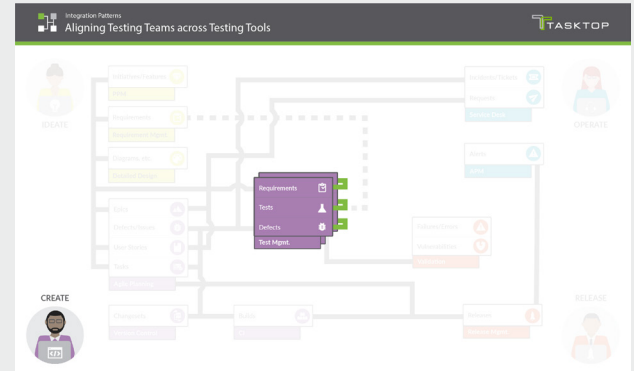
About

More often than not, single organizations will deploy multiple Agile planning tools (and/or instances thereof). This might be due to team preference, legacy tooling, technology needs or Agile methodology. Whatever the reason, teams still need to collaborate and managers need cross-tool visibility on development activity and performance.

This integration pattern synchronizes implementation artifacts between two or more Agile planning tools. The flow may be fully bi-directional, meaning all artifacts are replicated fully across all tools, or it may be more directed – replicating artifacts from multiple sources into a single tool.

Why

- Allows Agile teams to collaborate across tool boundaries
- Creates full traceability of related work across tools
- Allows flexibility for each development team to choose the Agile planning tool that best meets their needs



PATTERN 8

Name: Aligning testing teams

Stage: Create

What: Brings all the Test and QA teams together for cross-team reporting on product quality.

What's flowing?

Test Items
Requirements
Test Cases
Test Results
Defects

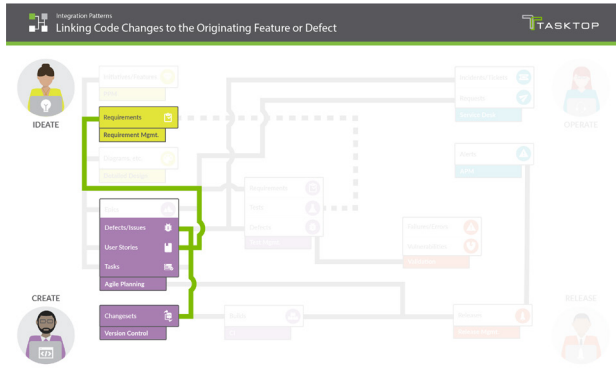
About

With the introduction of automated testing and the advances in testing tools, organizations often use multiple testing tools for their QA purposes. One tool may be used for test management, test execution, defect tracking and reporting, while another is used for automated testing.

This integration pattern flows information between the testing tools to create a consolidated view of defects for collaboration, reporting and analysis.

Why

- Improves product quality by seamlessly supporting the use of multiple specialized testing tools
- Enables cross-team reporting on defects



PATTERN 9

Name: Code changes to original feature/defect

Stage: Ideate and Create

What: Ensures all changes to a product during development are linked to the original request.

What's flowing?

Code changes	Implementation Items
Changesets	Requirements
	User stories
	Taks
	Defects

About

When dealing with millions of lines of code, it's very important to know why the code was written in the first place. That means linking the code changes to the originating feature or defect. However, doing so manually is slow, tedious and error prone.

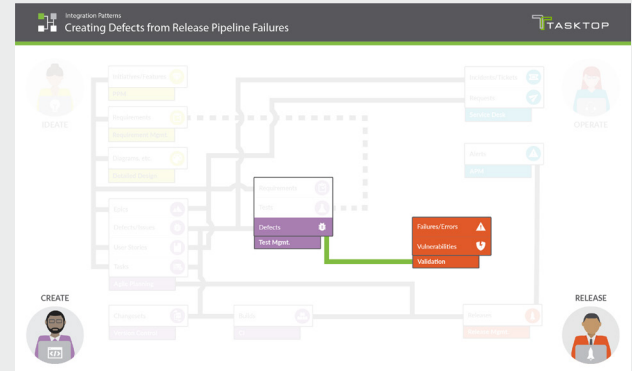
This integration pattern automates code traceability by automatically linking software requirements and defects to the lines of source code that implement them.

Two flavors exist for this pattern:

- Upon code commit, update an existing artifact (requirement, user story, defect or task) with the link to the changeset
- Upon code commit, create a new artifact for the changeset and link it to an existing artifact (requirement, user story, defect or task)

Why

- Ensures every line of code is linked back to original feature or defect
- Crucial for safety-critical lifelong products (such as airplanes)



PATTERN 10

Name: Defects from release pipeline failures

Stage: Create and Release

What: Flagging issues in release to developers and test teams.

What's flowing?

Release items	Implementation Items
Vulnerabilities	Defects
Failed tests	Bugs
Failed code scan	
Performance error	

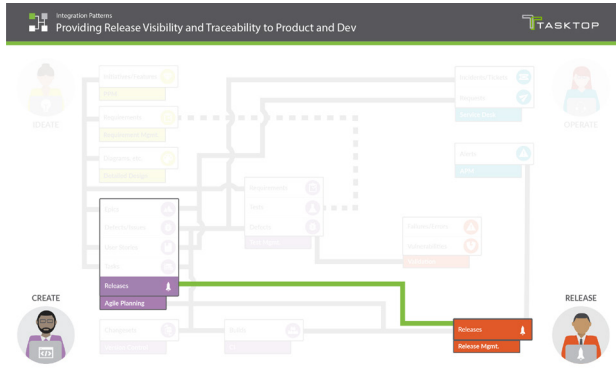
About

Before releasing software into production, many DevOps tools may be triggered (manually or automatically) to ensure code quality and prevent the destabilization of a production environment. A good example of that is scanning for security vulnerabilities with a tool like WhiteHat Sentinel or a crash from a tool like New Relic.

This integration pattern takes errors generated by those tools during the release process and automatically creates them as defects in the Testing tools or Agile Planning tools, so they can be immediately addressed.

Why

- Resolves failures faster to minimize impact
- Improves compliance by providing traceability between defects (e.g., security vulnerabilities) and fixes



PATTERN 11

Name: Release visibility and traceability to product and Development

Stage: Create and Release

What: Flowing release activities upstream.

What's flowing?

Release Items	Ideation Items	Implementation Items
Release	Release	Release

About

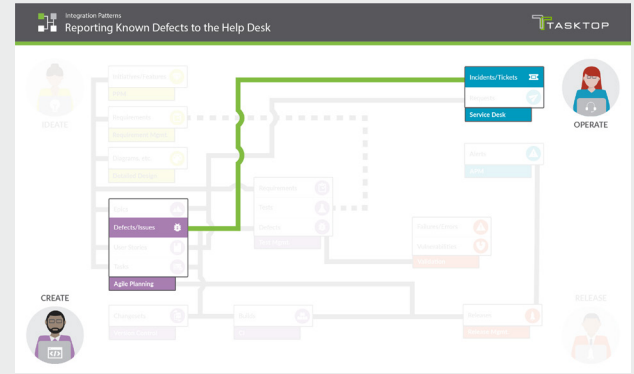
Release information varies across tools. In some tools, the release itself is a first class citizen; in others, the release is a field on a work item.

This release pattern makes it easier to designate one tool as the source or “master” of the release artifact, and use synchronization to populate the release artifact to other tools to be consumed there.

It mirrors a release created in one tool, for example, a Release Orchestration and Automation tool, and synchronizes it (with name, description, start and end dates, status and other fields) into other tools, for example, Agile Planning tools. As the release progresses, its status is visible in all tools.

Why

- Release information is populated automatically from the source to other tools
- Creates visibility into the release status for specialists who don't normally use Release Orchestration tools, like Product Owners, Project Managers and Developers
- Creates traceability between artifacts and releases by flowing the information through Tasktop Integration Hub's models



PATTERN 12

Name: Defects to the help desk

Stage: Create and Operate

What: Enables the people who build the software to update the people who work closely with the customer of any known issues (defects) going into production.

What's flowing?

Development Items	Testing Artifacts
Defects	Incidents
Issues	Tickets
Bugs	Problems

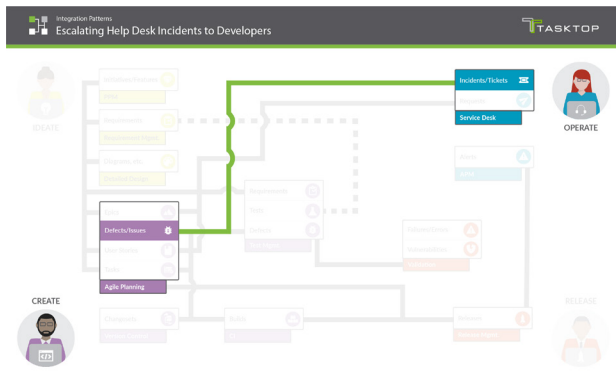
About

When a new release of an application is put into production, it is frequently the case that the development team was not able to address all known defects. However, in order to provide excellent service to its customers and to avoid triaging an incident only to discover later that it is a known problem, the help desk must be made aware of these defects.

This integration pattern mirrors defects from developers' defect tracking systems (often Agile planning tools) directly to the help desk software. Any updates are bi-directionally synchronized. For example, if additional information about the problem is reported to the help desk, this information can augment the defect report. Similarly, if progress is made against fixing the defect, that information is immediately available to the help desk.

Why

Customer issues are fixed faster



PATTERN 13

Name: Help desk incidents to developers

Stage: Create and Operate

What: Brings the people who log customer product issues closer to the people who fix them.

What's flowing?

Service Desk Items	Implementation Items
Incident	Bugs
Problem	Defects
Ticket	Stories
Issue	Issues

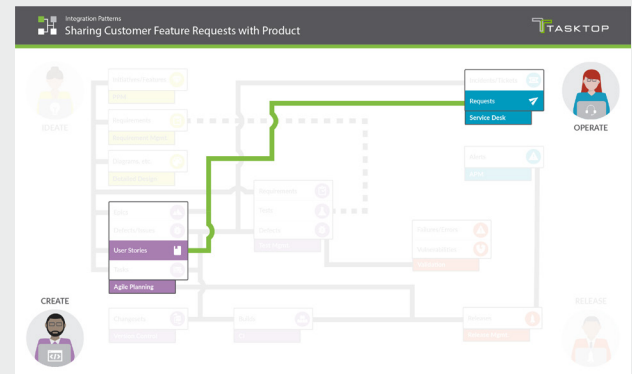
About

Many IT support teams are frustrated with duplicating ticket information from tools such as ServiceNow to other tools. Keeping different systems of record up-to-date through emails and IMs wastes precious time, and undermines collaboration between IT staff and developers.

This integration pattern eliminates those issues by creating a completely visible and traceable defect reporting and resolution process. Information automatically flows across ITSM and Agile planning tools. This pattern also allows for nuance; if the support engineer can discern whether the incident is truly a bug or a customer change request, bugs can flow to the developers as defects and change requests can flow as user stories.

Why

- Streamlines incident escalation processes, while providing transparency to the support engineer as work on the issue progresses
- Generates insight into the contributing factors to your true MTTR and identify the bottlenecks slowing down resolution times



PATTERN 14

Name: Customer feature requests to product

Stage: Create and Operate

What: Brings the people who need the features closer to those who plan and build them.

What's flowing?

Operation Items	Design Items
Customer request	Requests
	Requirements
	Features

About

Prospects and existing customers frequently raise requests to your customer-facing teams, including things like product enhancements. This pattern automates the flow of those requests from the business or support teams, working in a CRM like Salesforce or an ITSM tool like ServiceNow, to the product teams.

It connects a customer request with a product request, feature or requirement in an Agile Planning tool. It's also possible to flow the request to the tool Product Managers to use like a Requirements Management tool. As work on the request progresses, the synchronization ensures business teams have visibility on its progress and can communicate accordingly with their customers.

Why

- Product teams gain seamless visibility into customer needs without any manual work or correspondence outside CRM
- Sales and account teams can manage all the information about the account and opportunity in a single location

How Tasktop helps

Tasktop connects the network of best-of-breed tools and teams that plan, build and deliver software at scale. Our pioneering Tasktop Integration Hub addresses two core issues facing enterprises today:

- › **Software delivery work is invisible knowledge work. There are no physical materials to observe as they move through the value stream. It's hard to comprehend something you can't see, and even harder to manage it.**
- › **Unless fully automated from ideation to operation, transitions between key stages of the process are informal and untraceable.**

By connecting your complete value stream, Tasktop:

1. Automates the flow of information across the value stream

Enables the frictionless flow of artifacts (such as defects, user stories, trouble tickets), as well as information from events (such as build failures, version control changesets, security scan vulnerabilities and performance monitoring alerts), across the tools and stakeholders in the software development value stream. This removes non-value added work and bottlenecks; increases velocity and capacity; enhances collaboration; enables automated traceability and even improves employee satisfaction.

2. Enables visibility into the value stream

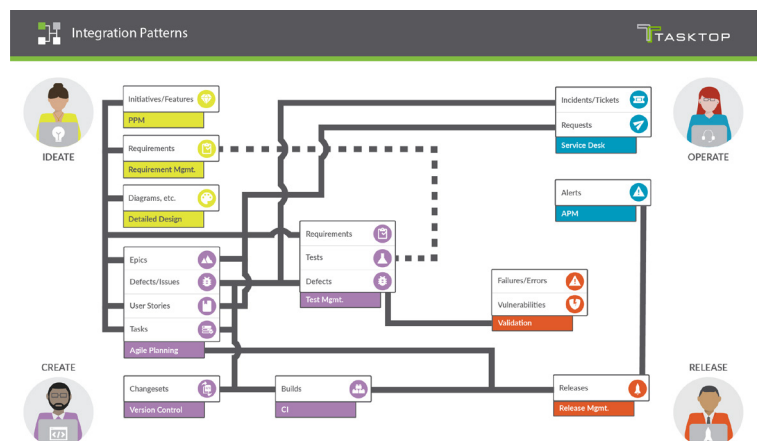
When managers want to see metrics and dashboards to understand project status, to optimize the process or to ensure compliance, it has been nearly impossible to get a real-time, holistic view across unintegrated tools. Tasktop unlocks lifecycle data from these application tool silos by automatically compiling lifecycle activity data into a single database. This data can be used to create consolidated, full-lifecycle reports and dashboards, as well as for traceability reporting.

3. Creates a modular, Agile toolchain

Software innovators require a best-of-breed tool strategy. Tasktop enables organizations to use the products that best support each discipline while getting the benefits of a single, integrated toolchain. Drives more value from each tool; allows organizations to easily add, replace and upgrade them, creating a proactive environment for innovation.

Getting Started

You should identify the scenarios that are causing the most immediate pain and enable the associated integration pattern. Over time, you should look to adopt more of these foundational integration patterns, linking them together to accelerate flow between the Ideation, Creation, Release and Operation stages that continuously plan, build, deliver and support software at scale. Once all tools are connected from end-to-end, you have a traceable and visible workflow that enables you to begin reporting and measuring performance through powerful Flow Metrics.



Flow Metrics are a unique set of metrics, per product value stream, that measure the rate of value delivery as correlated to business outcomes (business value, quality, improvement of software architecture and operational infrastructure, risk).

Measuring Your Flow

Flow Metrics are the Holy Grail for any organization looking to deliver more value through software. They offer business and IT leaders real insight into the health of their IT organization. What's working. What's not. What the pebble in the shoe is. Where to invest resources to optimize the process and innovate faster. Flow Metrics hold the secret to accelerating value delivery through software. Tasktop, as the leader in enterprise-grade toolchain integration and Value Stream Management, can help you connect your integration patterns and access your Flow Metrics.

Let's chat

Arrange a complimentary one-hour demo to see how Value Stream Integration can begin solving your acceleration issues almost immediately.

REQUEST CONSULTATION



ABOUT TASKTOP

Tasktop connects the network of tools, teams, disciplines and processes required for planning, building and delivering software at scale.

The backbone of the most successful Agile and DevOps transformations, Tasktop is an easy-to-use, scalable and reliable tool integration infrastructure that helps enterprises connect, visualize and measure software delivery value streams.

With the ability to support hundreds of projects, tens of thousands of users and millions of artifacts, Tasktop automates and traces the flow of work from customer request to finished software product.

For more information about Tasktop, please visit tasktop.com



Discover More



Value Stream Management and Value Stream Mapping - A complementary approach to delivering greater value through software



Beyond CI/CD: why you need Value Stream Integration alongside Application Release Automation



Why models are the only way to scale software delivery toolchain integration

info@tasktop.com

+1 778-588-6896

 tasktop.com

 [@tasktop](https://twitter.com/tasktop)

 TASKTOP

© 2007-2019 Tasktop Technologies Incorporated. All rights reserved. Tasktop and the "Less is More" logo (<=>) are registered trademarks of Tasktop Technologies, Inc. All other trademarks are the property of their respective owners. TSWP-5.19