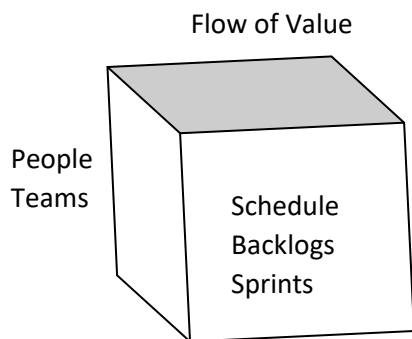


A Flow Perspective Focus on Value Delivery

Ken Pugh (ken@kenpugh.com)

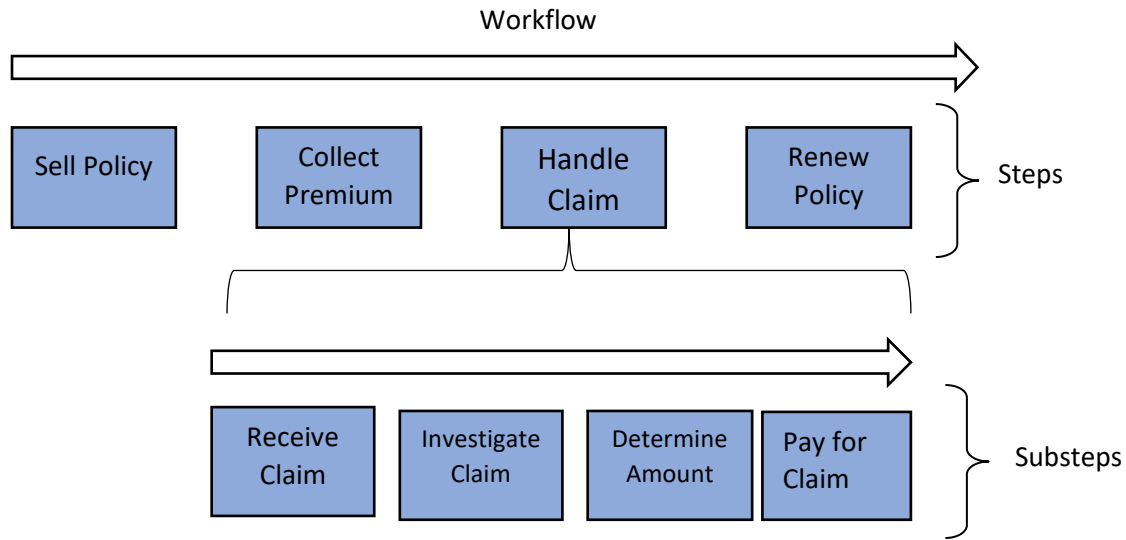
Introduction

Some agile practice articles discuss how individuals should be organized into teams and work collaboratively. Other articles describe how product backlog items/features and sprint backlog items/stories/tasks should be estimated and scheduled. In this article, the flow perspective focuses on how customer or business value is delivered. This flow is sometimes shown with value streamsⁱ, story mapsⁱⁱ, customer journeys, workflows, or other means. Understanding and making the flow visible can help with organizing the other views of agility. A more detailed examination of the relationship of flow to other aspects of agile development will follow in a separate article.



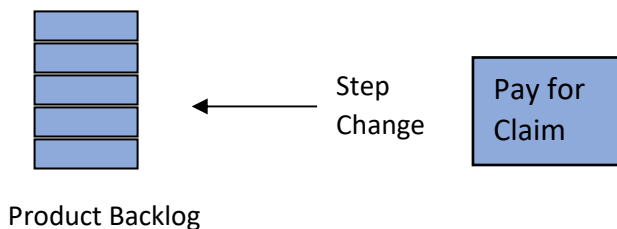
The Example Flow

The insurance industry has a common flow. A simplified version of the flow is below. Having a shared understanding of this flow helps collaboration. Most flows have a hierarchy of steps, as shown by the steps beneath Handle Claim.



Flows may be linear, where one step progresses to the next. There also may be loops, where a step is repeated. If there were multiple claims, the Handle Claim would be repeated. Steps may have alternatives. Handle Claim may work the same way for the first two claims and then work differently on the third claim.

A step in the flow (either at the highest level or at lower levels) can correspond to a product backlog item or feature. Alternatively, all the steps in a flow may be represented as a single item in a product backlog.



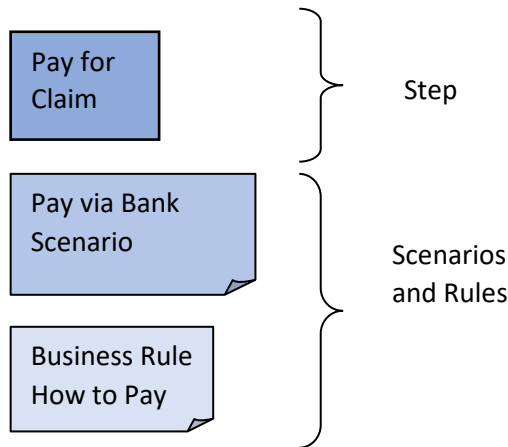
Example Scenarios

Steps can be detailed with scenarios that describe in more detail how a customer or user interacts with the system. A common template for a scenario is:

```
Scenario: Title
Given some conditions
When the user performs an action or an event occurs
Then an output is generated and/or some condition changes
```

The scenarios can be created by the Triad (Customer, Developer, Tester). The Triad represent three views of the scenario – the requirement details, how the requirement may be implemented, and critical analysis of the requirement and the implementation.

Each step has at least one, if not many, scenarios associated with it. The scenarios may also refer to business rules that further define the scenario. For example:



A scenario for Pay for Claim via Bank might look like:

```

Scenario: Pay for Claim via Bank
Given amount for claim has been approved
And insured has a bank account
When the claim is paid
Then a payment is made via our bank
  
```

Here's an example of a business rule that determines which scenario to use to pay a claim:

```

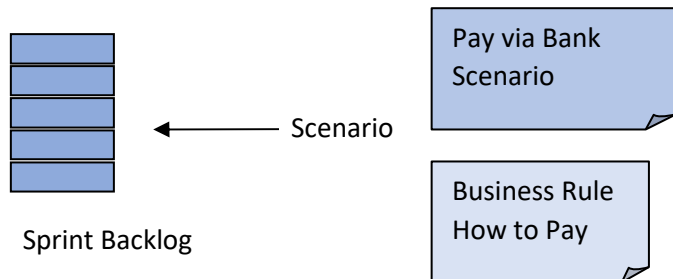
Scenario: Business Rule for How to Pay Claim
** Claim is paid via different methods
| Insured Payment Information | Payment Method |
| US bank account            | ACH             |
| Non-US bank account        | Wire            |
| No bank account            | Check           |
| Paypal account              | Paypal          |
  
```

Scheduling

One scenario or one business rule can constitute a sprint work item / story. Alternatively, a group of related scenarios can be part of a work item. If there are only a few scenarios in a step, the entire step could be the work item. Work items automatically have a context – the steps in which they appear.

Defects are often related to a scenario. If they become a story/work item, then they should reference the applicable scenario or step. Quality attributes (non-functional requirements) for a scenario, such as performance, can be part of the story for that scenario or a separate story.

Enablers/development items are often tied to particular scenarios and should be placed at the appropriate place in the backlog. For example, there could be an enabler that connects to an actual ACH gateway and thoroughly tests that connection.



Scenario Expansion

A scenario can be explored by the Triad before putting it onto a backlog or after pulling it off a backlog. The exploration can expand the scenario with data that helps further shared understanding. With data, the previous scenario might look like:

```
Scenario: Pay for Claim via Bank
Given amount for claim has been approved
| Amount | Insured |
| $100   | Sam     |
And insured has a US bank account
| Insured | Account Number | Routing Number |
| Sam     | 1234567890     | 987654321     |
When the claim is paid
Then a payment is made via our bank to
| Account Number | Routing Number | Amount |
| 1234567890    | 987654321     | $100   |
```

Scenarios and Tests

The expanded scenario represents the details of the interaction. It also serves as a test. When it is executed, the condition listed in the Then part must happen. If the check does not pass, the test fails.

```
Then CHECK a payment is made via our bank to
| Account Number | Routing Number | Amount |
| 1234567890    | 987654321     | $100   |
```

Automation of tests allows for rapid checking that new scenarios do not cause issues with previously implemented ones. These tests are independent of the implementation. The tests can be run as part of continuous integration.

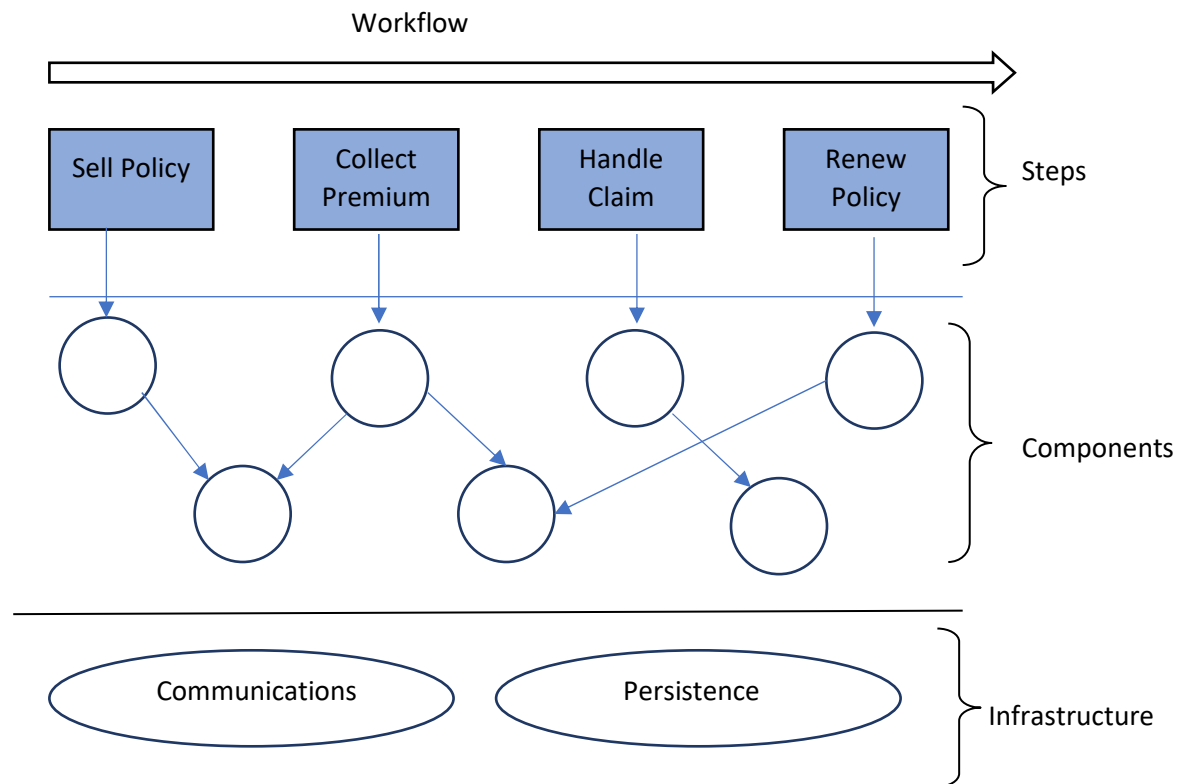
The scenario can be used during sprint demo to show that the application successfully executes the requirement.

Exploratory and usability testing are also needed to see that the scenario does not cause unexpected or unspecified conditions to occur and that the user interface is usable.

Implementation

The workflow is implemented with components – software and hardware. The software components might be classes, modules, microservices, or applications. In any case, they must execute the specified flow with the specified quality attributes. One way to achieve lower impedance between the flow and the components is to have components that directly relate to the flow.

Most components need an infrastructure in which to execute. The decisions on the component breakdown and the infrastructure are technical and may change as new scenarios are implemented or new technology is developed. Passing all the tests for the scenarios indicates that the new implementation or technology still meets the requirements.



Team Responsibility

Scenarios that are part of a step often have more cohesion with each other and less coupling to scenarios that are part of other steps. If a workflow requires more than one team, assigning responsibility for scenarios based on the steps they represent may reduce the inter-team dependencies that need to be resolved.

Summary

A flow perspective ties together the delivery of customer and business value with its composition of scenarios. User stories consist of one or more scenarios or business rules. The detailed scenarios form the functional tests for the workflow. Automation of these tests allows for rapid checking that implementation changes have not affected the delivery of customer and business value.

ⁱ Mary and Tom Poppendieck

ⁱⁱ Jeff Patton