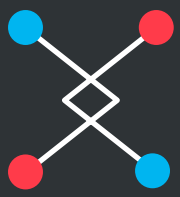


MIGRATION SERVICES

# Vaadin Swing Migration Service

Unlock the value of existing applications. Un-burden your developers.

vaadin

**Planning to complete a migration of your Swing desktop application to Web?** Vaadin Components, the Flow framework, and Pro tools are the best way to get your developers productive with HTML5, web components, and the cutting edge of progressive web applications.

**But how do you get there?** If your developers are currently maintaining a legacy application and tailoring functionality to new user requirements, they will be heavily burdened if they need to continue this work while reskilling on web technologies and developing a brand new product from scratch at the same time.

The Vaadin Swing Migration Service can help you accelerate your migration goals while freeing your developers to focus on the things that matter most - delighting your existing customers, preparing to support new customers, and helping existing customers make the transition from old to new.

Vaadin helps companies complete their migration plans successfully with a three-phase approach consisting of preparation, the migration itself, and hypercare.

# Preparation

To ensure the project gets off on the right foot, there are various exercises we do before the migration starts. Typically these combine the top-down perspective on the one hand of analyzing stakeholder needs and organizational limitations, with the bottom-up perspective of technical complexity and the volume of code. These two perspectives are treated in three components: the questionnaire, the optional Proof of Concept, and the Migration Assessment.

**First is the questionnaire.** The questionnaire is a document containing a number of general, high-level questions about the project. After completion, our experts will review your answers and extrapolate from similar case data to produce ballpark figures on the duration and effort of a full migration. This figure is non binding but should give you clarity on the rough order of magnitude companies with similar application setups have invested.

A more concrete way to prepare the project technically is the **POC (Proof of Concept)**. The POC is optional and it should be seen on a case by case basis if it is required. There is no real set duration for a POC because it will depend on the complexity and number of concepts that need to be proved. The product of the POC is typically a piece of working Vaadin software.

Executing the POC will give you a number of lessons learned that in turn give insight to the challenges you would face in a full project. These are related to the technical obstacles and their resolution, the experience of combining your developer team with a team of Vaadin consultants, and the duration of the exercise.

If you are required to write an RFP for the full migration project, the information you get from the POC will give you experience data to make this document useful.

The final preparation component is the **Migration Assessment**. With the technical data we've gathered in the POC, the questionnaire, and from past experience, we're ready to start planning. We will review the full source code of your application and potentially conduct a number of interviews with developers, support teams and other stakeholders. As we get the full project into scope, we can discuss plans with you for ways to divide up the work and track progress in a formal project.

**Migration Assessments typically cover four topics:**

- Technical impact: how the migration will impact the code of the existing system,
- Collaborative models: how the work will be organized and how your team will work with Vaadin consultants,

- Equivalence constraints: how we will achieve optimal reuse of the existing application and for which parts we will switch to a development mode, and
- Acceptance criteria: how we will test the new system and determine the “definition of done”.

What you can expect from a Migration Assessment as concrete deliverables would be agreed beforehand, but would typically include project materials like WBS tables, gantt charts, milestone descriptions, commercial proposals for specific work topics, or technical data including static code analysis results, custom and core component or API mapping specifications, or descriptions of which software transformations would be candidates for automation.

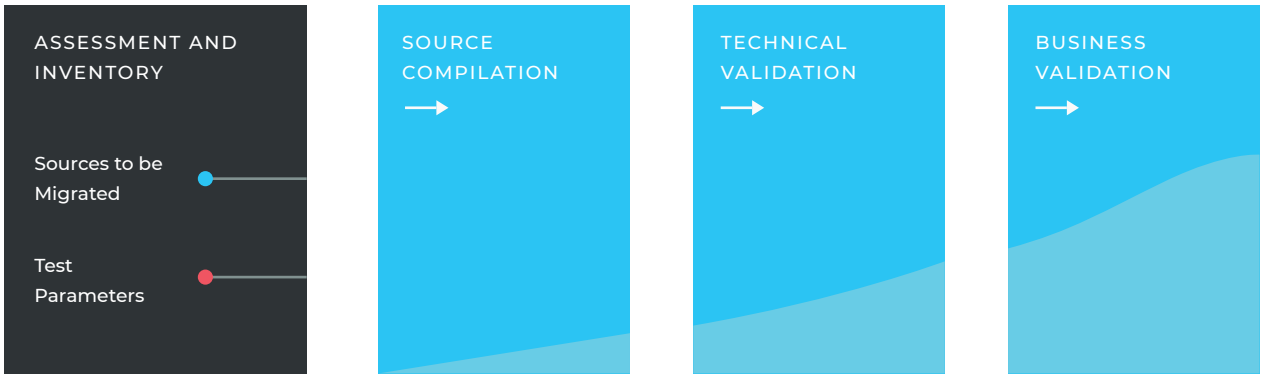
# Migration

Migration starts with a kickoff and ends when the new software is accepted and begins to be used instead of the Swing application. In technical migrations, especially those in which the transformation rules are formalized, the flexibility is the highest to determine the moment at which developers terminate maintenance of the Swing application and begin maintenance of the new.

With technical migrations, the project will mostly consist of a longer preparation time during which the transformation tools are adapted to suit the specific needs of the project (think custom components, third-party libraries and integration with peripheral devices) and the new software is tested, while the maintenance of the legacy application continues uninterrupted.

A formal approach we use in technical migrations is to iteratively follow a series of progressively comprehensive testing exercises based on snapshots from the Swing application. If the existing software is being maintained, we will align the cadence of the project iterations with the Swing application's release schedule.

With each iteration, we take a snapshot of the actively maintained application, apply transformation rules and finish with manually merged spot fixes. Progressively comprehensive testing then happens for design-time, functional runtime, and non-functional runtime aspects. Project



progress then can be measured against milestones that correspond to key tests in each of these areas succeeding.

In addition to the transformation of code, relevant integration and runtime topics including desktop integration with peripherals, data migration and multitenancy, security, and scalability are handled. Finishing services like developer training, ops training or documentation will typically be held immediately before acceptance, as part of the acceptance protocol.

# Hypercare

Your teams may find it useful to include hypercare and maintenance in the migration service package, or arrange for support with a Pro subscription. These will ensure extra support during the first months and years after the migration completes and access to a host of other benefits.