



ELTONIC Development Process

Quick Guide

Every project is unique

You know your business inside out. You have thought and planned your ideas carefully and are keen to see it live as soon as possible.

You have prepared a project brief, listed specific objectives and requirements, mocked up application screens and have even identified a few design ideas. You're now ready to start the development process.

This is where we come in.

Agile Development

Whether you're focusing on developing web applications, mobile applications or enterprise multi-platform applications, agile seems to be everywhere. But what is agile exactly?

In simple terms, it's a methodology that reduces development cost and risk, improves quality and enhances customer's satisfaction by better meeting project's needs and expectations.

We use agile methodology in our process, which is a proven path to ensuring the projects deliver better.

In more Agile projects, approaches which we have found successful are:

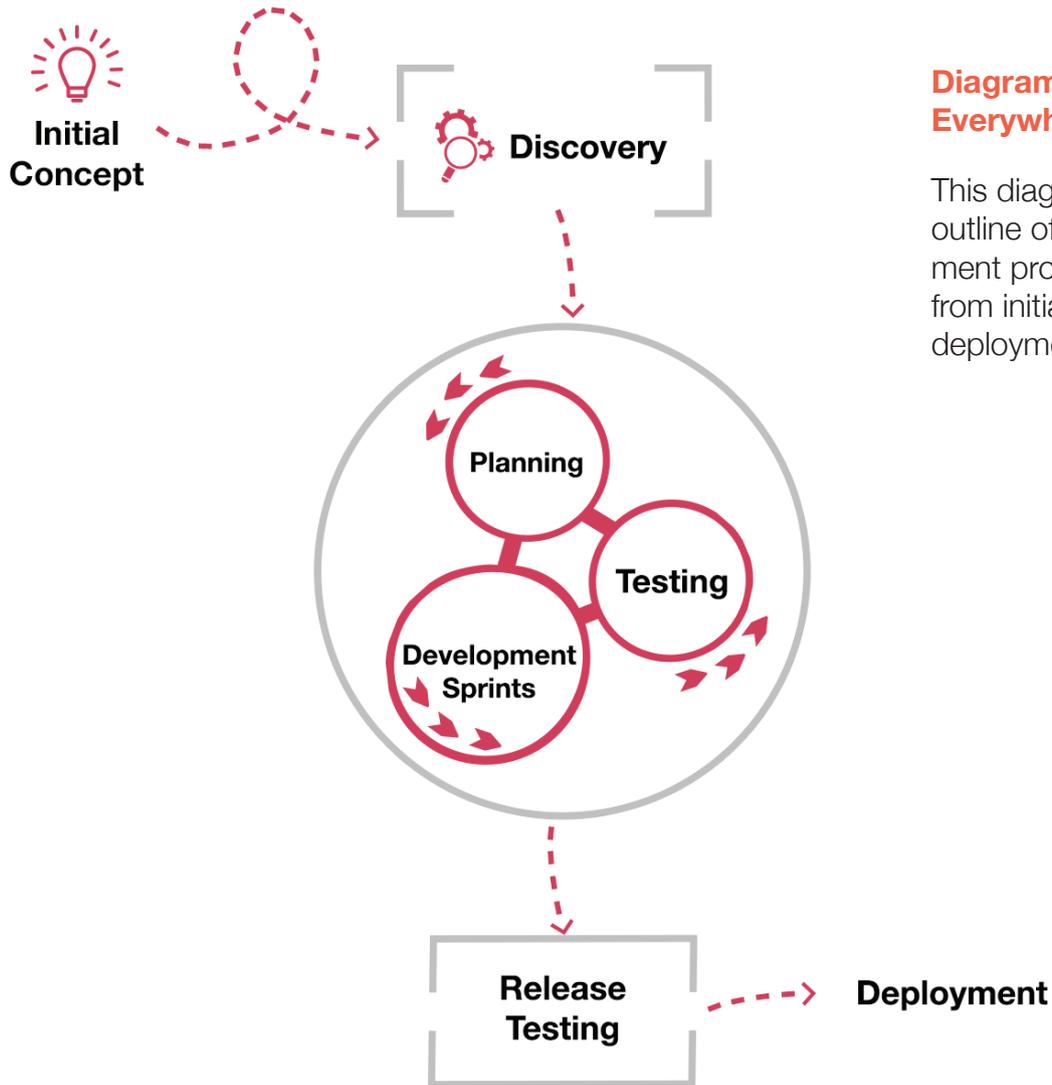
- Close, **daily cooperation** between business people and developers
- **Changing requirements** are welcome, even late in development
- Working software is **delivered frequently** (weeks rather than months)
- **Working software** is the principal measure of progress
- Continuous attention to **technical excellence and good design**
- **Simplicity**—the art of maximizing the amount of work not done—is essential
- **Self-organizing** teams

Did you know?

Heathrow's Terminal 5 was an example of an agile project which was delivered on time and on budget – often compared to the build of the new Wembley stadium which was more than a year late and went over the original budget by more than 100%.



THE PROCESS



Diagrams and mockups. Everywhere.

This diagram shows an outline of the development process of a project from initial concept to deployment.

The discovery phase is undertaken at the beginning of the project. It will consist primarily of requirements gathering workshops, interviews with key stakeholders and users of the end product. The key output of this phase is the Backlog. This is a set of all the identified requirements written as 'user stories'.

User stories

A user story is a requirement, feature and/or unit of business value that can be estimated and tested, consisting of the following elements:

- **description**, usually in business terms;
- a **size**, for rough estimation purposes, generally expressed in story points (such as 1, 2, 3, 5)
- An **acceptance test**, giving a short description of how the story will be validated (see the example below)

Specific deliverables for the discovery phase include, but not limited to the following:

- **Backlog**, a comprehensive list of structured user stories, each estimated according to complexity
- **Technical solution documentation**, detailing parts of architecture
- A set of **wireframes** for specific use cases

Quick Agile Glossary

Backlog

A collection of stories and tasks the Sprint team will work on at some point in the future.

Sprint

A period (from 1 week to 2 months in duration) during which the Agile development team produces an increment of completed software.

Release

The movement of a software product or system from development into production.

Example User Story

Description:

As a purchaser on the website,
I want the ability to pay with a credit card,
So that I may immediately confirm my purchase

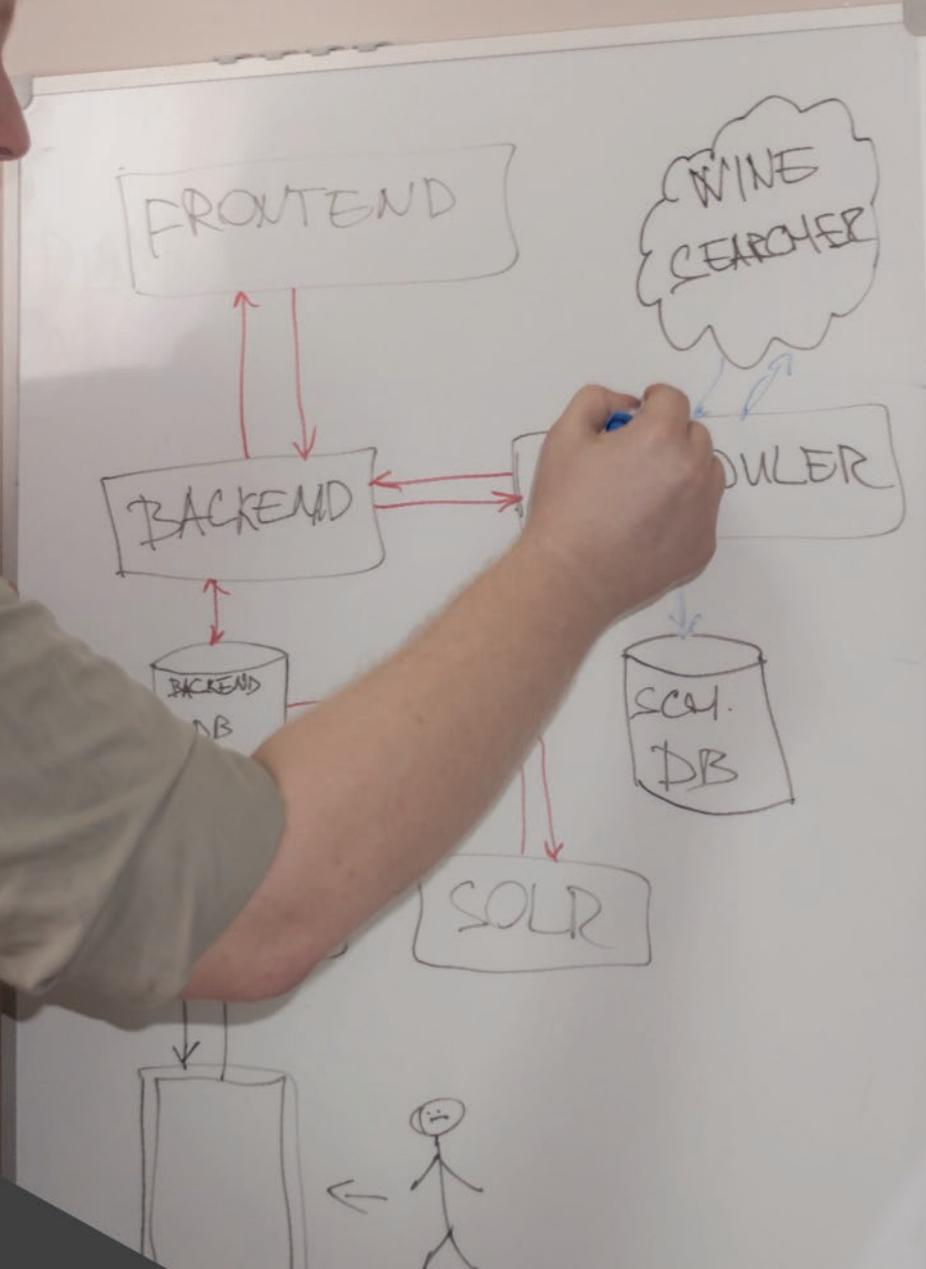
Acceptance Criteria:

Accept Visa, Mastercard, AE
Validate CC when entered
Validate expiration date and CW
Validate billing address
Generate success or failure messages after processing

Story Size



5



DEVELOPMENT SPRINTS

A sprint is a set period of time during which specific work has to be completed and made ready for review.

Planning

Each sprint begins with a planning meeting. During the meeting, the product owner (the person requesting the work) and the development team agree upon exactly what work will be accomplished during the sprint. The development team has the final say when it comes to determining how much work can realistically be accomplished during the sprint, and the product owner has the final say on what criteria needs to be met for the work to be approved and accepted.

Sprint Duration

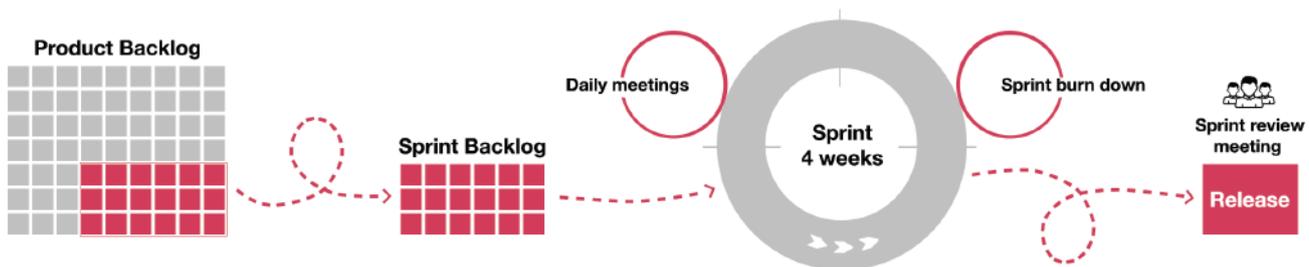
The duration of a sprint is determined by the scrum master, the team's facilitator. Once the team reaches a consensus for how many days a sprint should last, all future sprints should be the same. A sprint can last as little as 1 week and up to 30 days.

The kick-off

After a sprint begins, the product owner must step back and let the team do their work. During the sprint, the team holds daily stand up meeting to discuss progress and brainstorm solutions to challenges. The project owner may attend these meetings as an observer but is not allowed to participate unless it is to answer questions. The project owner may not make requests for changes during a sprint and only the scrum master or project manager has the power to interrupt or stop the sprint.

Completion

At the end of the sprint, the team presents its completed work to the project owner and the project owner uses the criteria established at the sprint planning meeting to either accept or reject the work.



TESTING & DEPLOYMENT

Agile Quality Assurance

Agile development recognizes that testing is not a separate phase, but an integral part of software development, along with coding. Agile teams use a "whole-team" approach to "baking quality in" to the software product.

Our testers are embedded into the team to work on automated acceptance testing right after the start of the sprint. Testing and coding are done incrementally and iteratively, building up each feature until it provides enough value to release to production.

Automated Regression Testing

Regression testing seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes have been made to them.

The intent of regression testing is to ensure that a change such as those mentioned above has not introduced new faults.

Where possible, we adopt automated testing to save time and ensure quality of delivered software.



Continuous Integration

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

By integrating regularly, we can detect errors quickly, and locate them more easily.

Release Process

In hosted software development, a development environment refers to a server tier designated to a specific stage in a release process.

You will hear us mentioning development, staging, and production environments.

Development Environment

Setup by developers, for developers. Usually carries the latest release of the code with testing data.

Staging

Mirror of production environment. Usually used as a final testing step before going live.

Production/Live

Serves end-users/clients



tel: +373 (22) 22 95 00
info@eltonic.com
www.eltonic.com

Copyright © 2016 Eltonic SRL.
All rights reserved. Some illustrations designed by Freepik.com.