



Get in touch with us:



Tech Radar - May 2015

Or how Obecto is staying fresh and current with new technologies and tools, while maintaining its focus on the industry standards. This is our May'15 edition of the Obecto Tech Radar.

We believe **software development** is more than a way to *deliver features...*

It should bring **valuable change** to the world

Introduction

One of the biggest challenges in our industry is how to stay fresh. New technologies, tools and practices are coming in all the time — especially new unproven ones. Technology strategy will be at the core of most business strategies, and it is essential to have a plan how new technology is adopted.

The Radar is one way to proactively seek for new ways to do our work and bring competitive advantage to our customers, while at the same time maintain focus on what we currently do as standard practices in our industry and maintain the Obecto portfolio of projects in production.

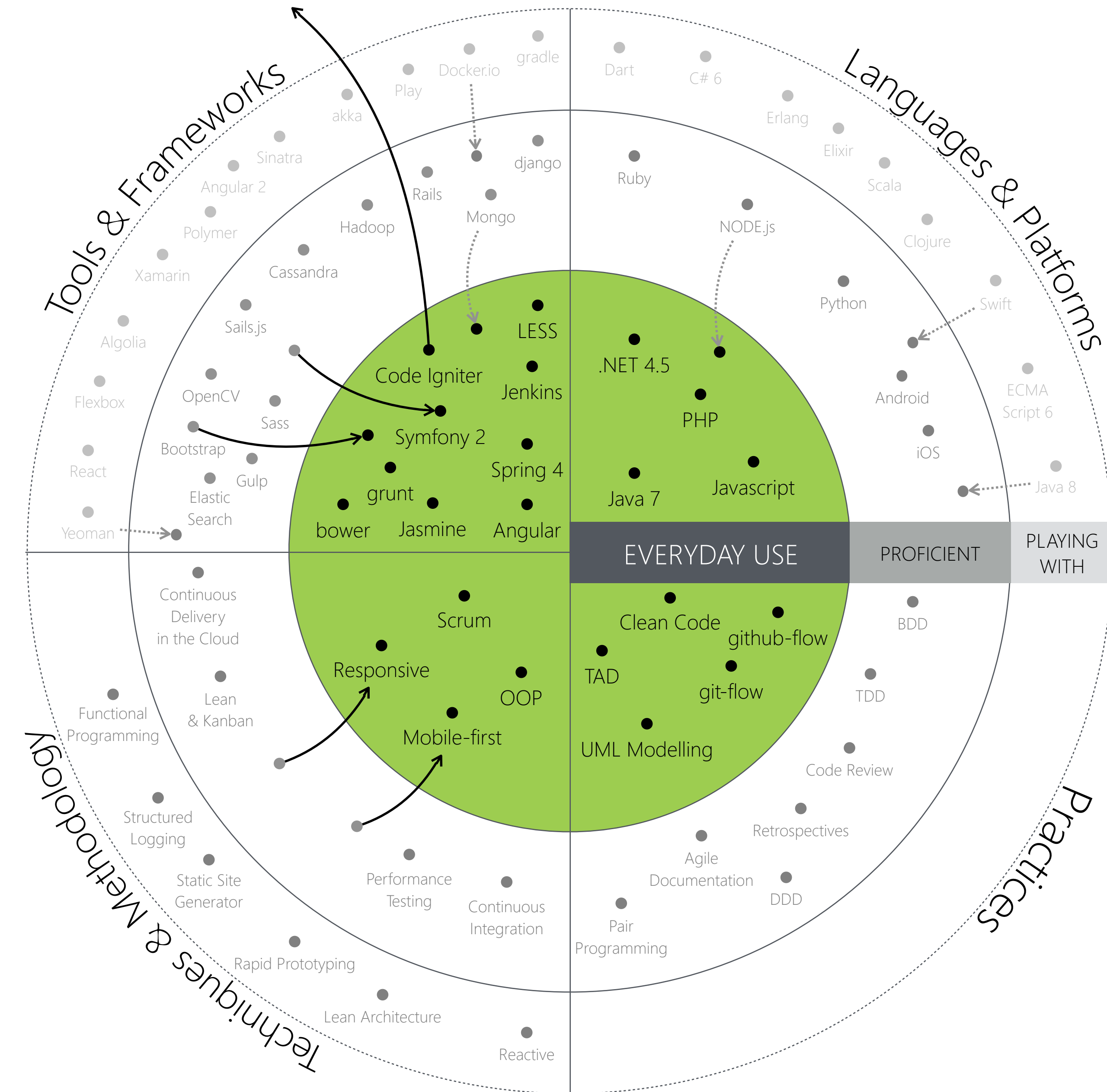
The radar is organised in 4 quadrants:

- **Languages & Platforms**
- **Tools & Frameworks**
- **Techniques & Methodology**
- **Practices**

Within a quadrant there are rings showing the stage of adoption of each point:

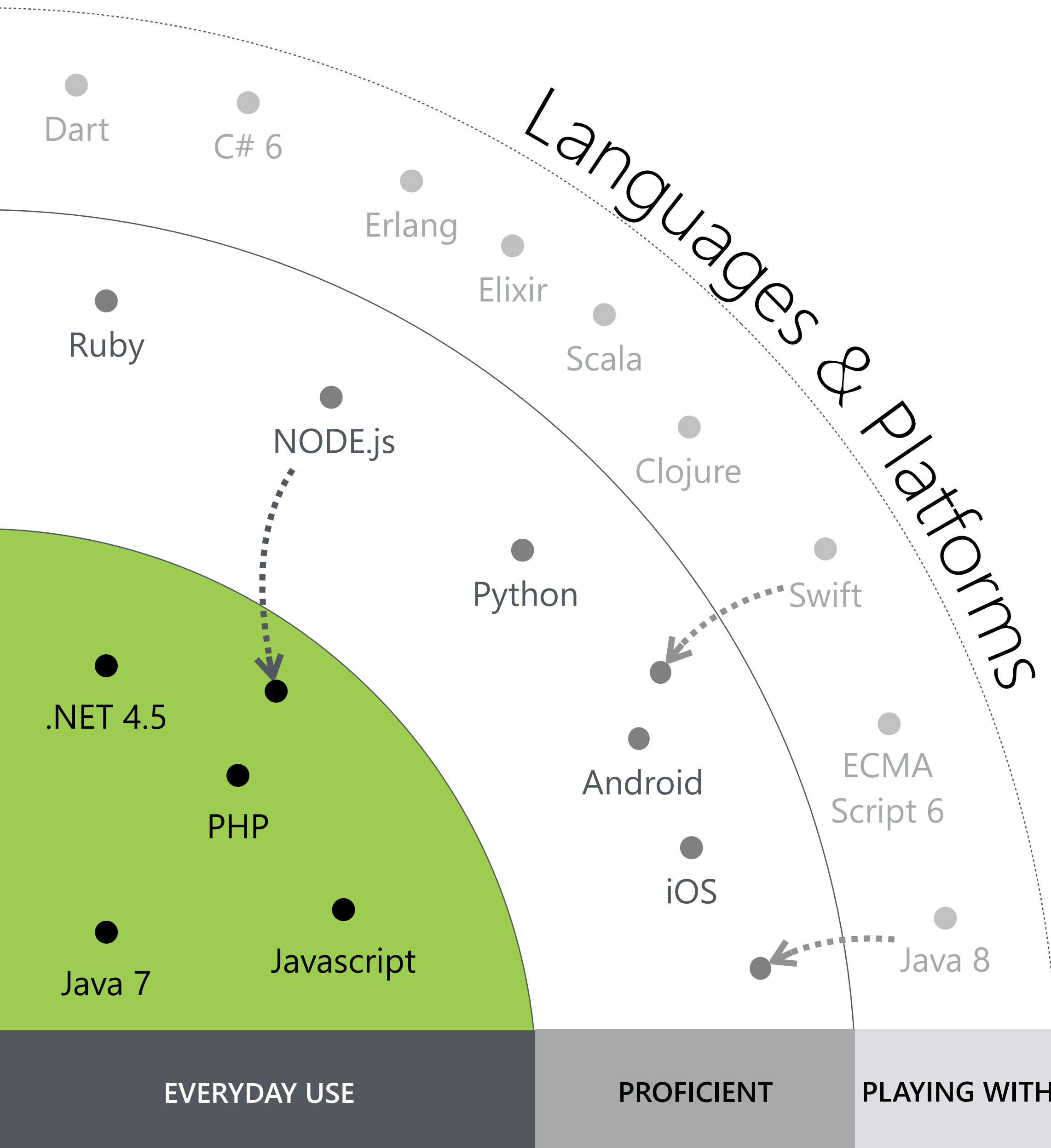
- **Everyday Use** — the final stage of adoption; considered to be our core expertise.
- **Proficient** — used in production projects, but yet to be fully adopted.
- **Playing with** — still in experimentation and research stage, used only in pet projects.

The following slides will focus on each specific quadrant.



Languages & Platforms

Our core expertise is in the following programming languages and platforms:



Some points from this quadrant that are worth highlighting:


- **NODE.js** — as our portfolio of NODE.js project grows, this platform is in trend to be fully adopted in the near quarters.
- **Java 8** — we're following closely the Java 8 readiness of all frameworks we currently use and plan to migrate existing projects to Java 8.
- **Android & iOS** — our native apps mobile experience portfolio is not impressive enough to consider this fully adopted. Nevertheless, we feel proficient in both mobile platforms. That's an expertise not to be underestimated!
- **Swift** — our next iOS project will be in Swift instead of Objective C.
- **Clojure, Scala, Erlang** — our team is quite excited about the functional programming languages. The following quarters will show if any new project with a solution that includes the functional style of programming.
- **ECMA Script 6** — we're experimenting with ECMA Script 6 to Javascript compilers that we can use in existing projects.

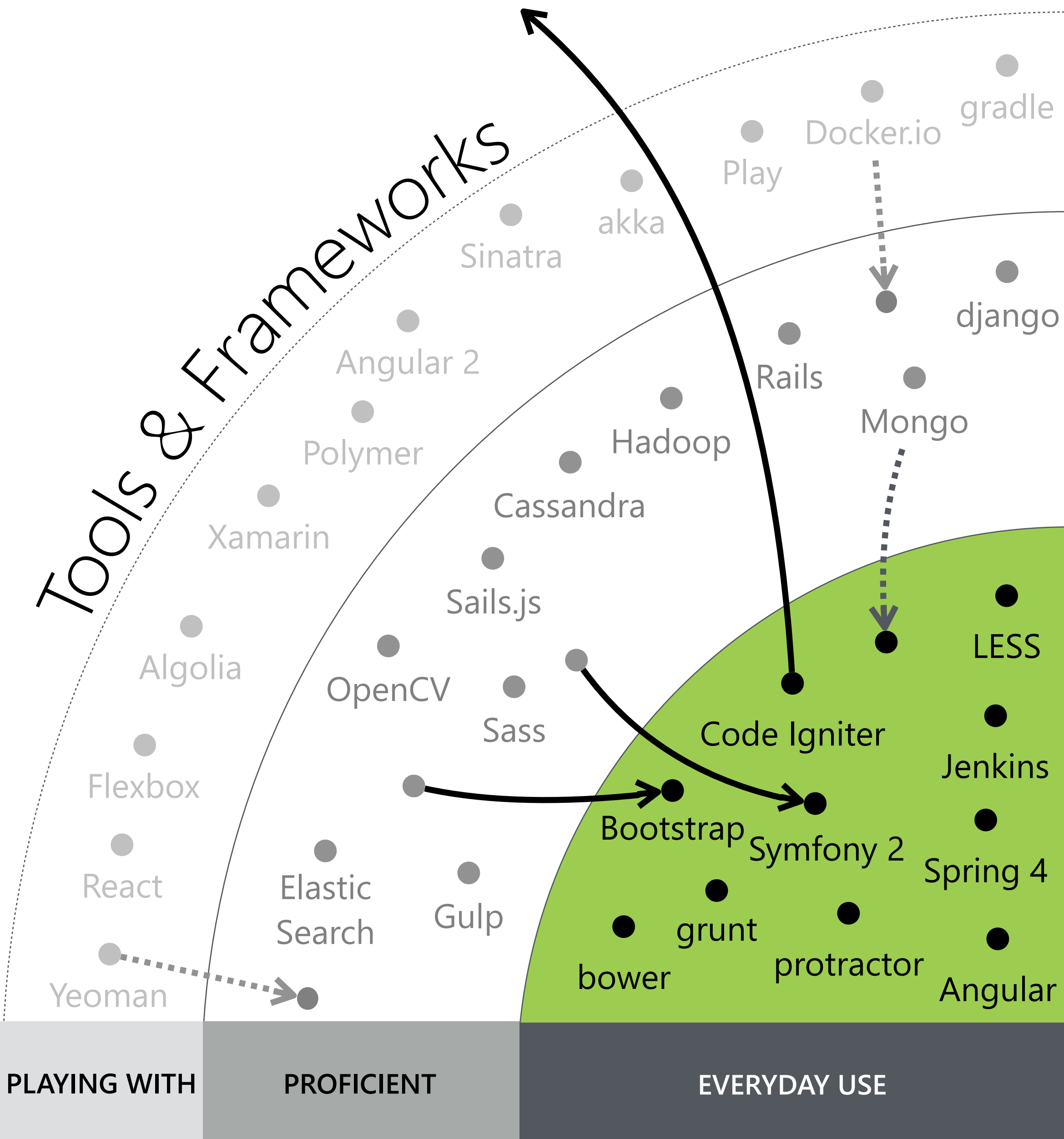
Tools & Frameworks

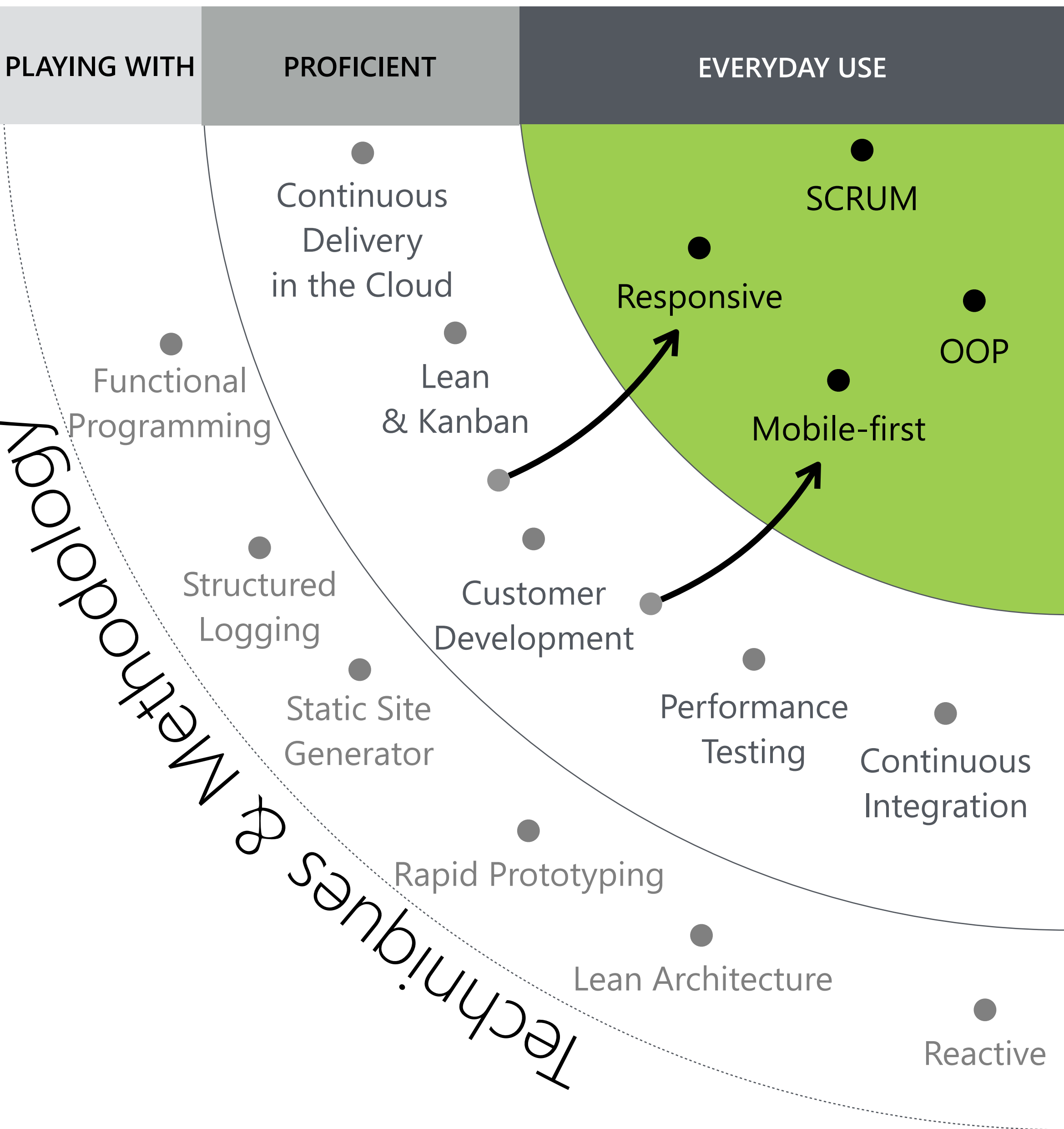
This is the most dynamic quadrant in our radar. Nevertheless, the stable core in the centre includes:



Our current selection of tools and frameworks is optimised for building very efficiently Single Page Applications (SPA) with backends providing REST APIs.

- **Symfony 2** — in the current edition of the radar we're putting on hold **Code Igniter** taking it out of everyday use. All new **PHP** projects are started with Symfony. All **PHP** projects in production have been migrated to **Symfony 2**.
 - **Bootstrap** — when it comes to *responsive* web applications and *mobile-first*, 2014 was a very busy year for us (*for more info check our mobile experience portfolio*).
 - **Mongo** — very soon Mongo will get its place in the core ring next to the relational databases we typically use (e.g. MySQL and PostgreSQL).
 - **Cassandra & Hadoop** — even though these are placed in the Proficient ring it worths mentioning that we have over 2 years of experience in using them as part of our analytics solution for **Newstag** (*for more info check our mobile experience portfolio*).
- 

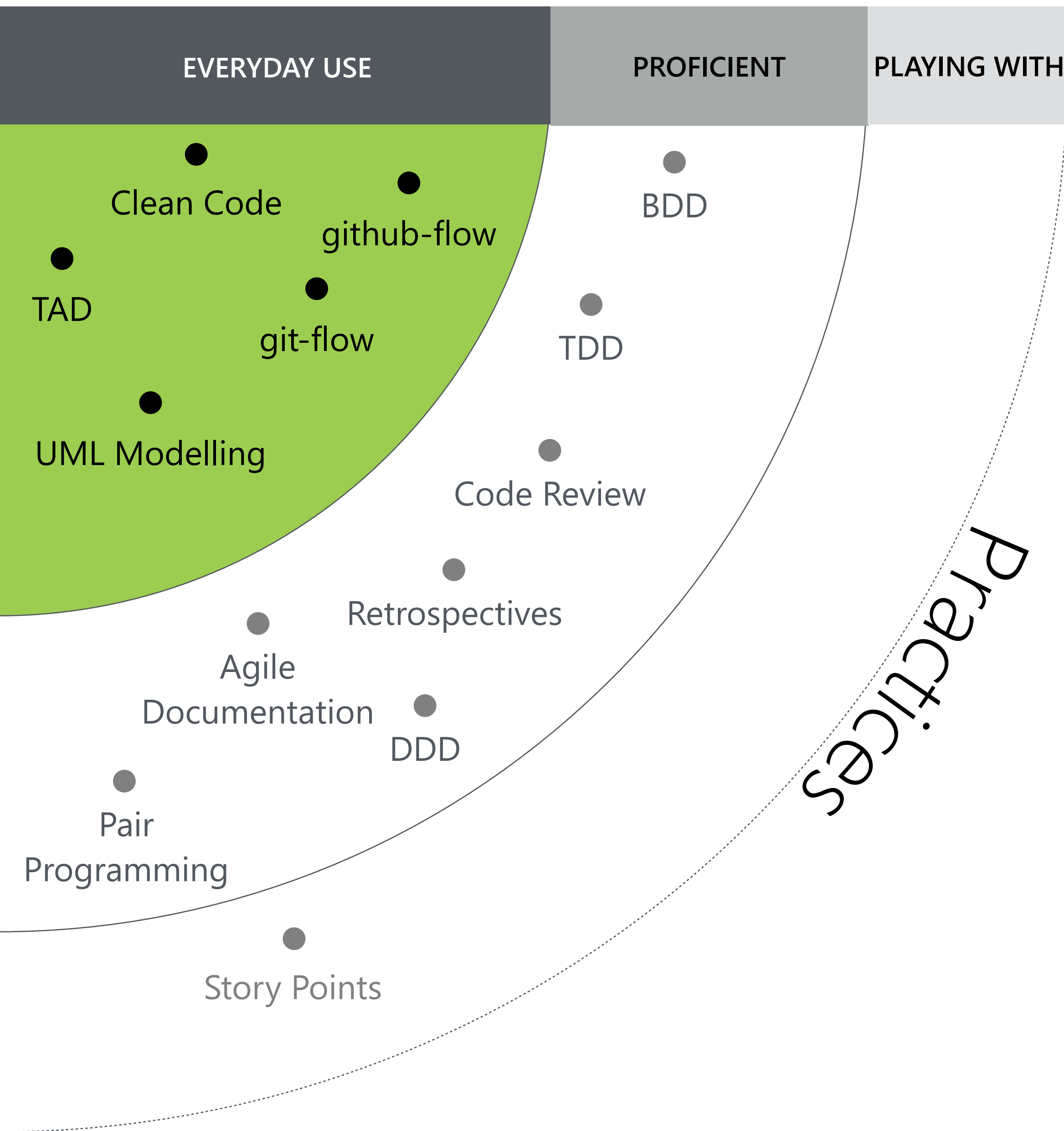




Techniques & Methodology

The process and methodology in Obecto has evolved to support projects of varying complexity and size with ever increasing focus on being as lean as possible.

- **SCRUM** — we consider ourselves experts in applying the SCRUM framework for projects that need to sustain longer term development.
- **OOP** — we still feel that the Object Oriented Programming paradigm when combined with good knowledge of the Design Patterns and the SOLID principles gives the most versatile approach for software projects.
- **Responsive & Mobile-First** — when it comes to *responsive* web applications and *mobile-first*, 2014 was a very busy year for us (*for more info check our mobile experience portfolio*). These two techniques have become a valuable additions to our core expertise.
- **Customer Development** — that's a lean startup methodology. Nevertheless, when applied in the context of the early stages of a software project it brings very efficiently validation and verification of the core concepts and assumptions by using a scientific approach.



Practices

Practices are specific elements that we use and apply in our software development process, such as how we write, or test, or release our code.

- **Clean Code** — we have projects that need to sustain development in the longer term. Cleanliness of the code is crucial when we need to maintain our productivity in the long term.
- **TAD** — Testing After Development is a practice where the unit test is written after the actual code, and only specific part of the code are put under such unit testing instead of being in pursuit of 100% code coverage. For many project we have done exactly the opposite and we're very proficient in **TDD** (Test Driven Development), as well as writing end-to-end tests with the help of **BDD** (Behaviour Driven Design).
- **Story Points** — we're currently playing with Story Points in the practice of estimation and we're looking forward to use this practice in our customers' projects.

Let's talk!

Yavor Popov | Account Manager
email: ypopov@obecto.com
mobile: +359 883 317 270
skype: yavor.obecto