

Humans in the Software Development Loop

July 24, 2023



Digital Experience Lead Research Analyst

Keith Instone has been practicing user experience for over 30 years, spanning academia (studying the science of human interaction with technology), industry (with IBM), and as a consultant (to startups, Fortune 100 companies, and in between).

Submitted by [Keith Instone](#) on 24, Jul 2023

Humans in the Software Development Loop



Humans in the Software Development Loop

Organizations continually demand that their software development teams deliver more functionality more quickly, and with higher quality. To address these challenges, high-performing teams rely on automation. And with [more](#) and [more](#) and [more](#) DevOps tool chain automation driven by machine learning and other [AI solutions](#), teams may feel the need to rush to try to automate everything. But before you get hyper about automating your software development life cycle, we recommend you first step back and do two things:

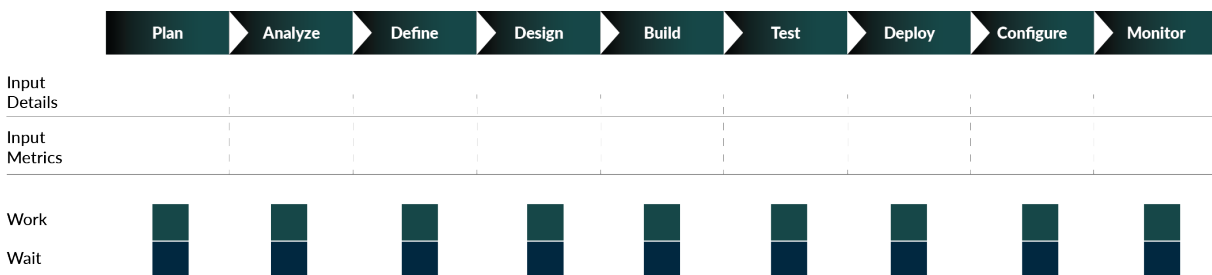
- Start value stream mapping as part of a value stream management program.
- Plan your developer experiences to identify which types of automation make sense at various steps in the process.

Value stream management

[Value stream management](#) (VSM) combines [people, process, and technology](#) to document and continually improve the flows that generate business value. VSM is touted as the next evolution of DevOps, but [there are still barriers to overcome](#).

Value stream mapping gives your business [metrics](#) that your automation efforts can help you collect. The [steps](#) for [value](#) stream [mapping](#) help you see the flow of work across your software development life cycle stages, including where time is lost and where value is added.

Gaining a complete, end-to-end perspective of your value streams allows you to improve throughput and quality while reducing waste. You can also focus automation on where it has the greatest impact for your customers.



A template for value stream mapping

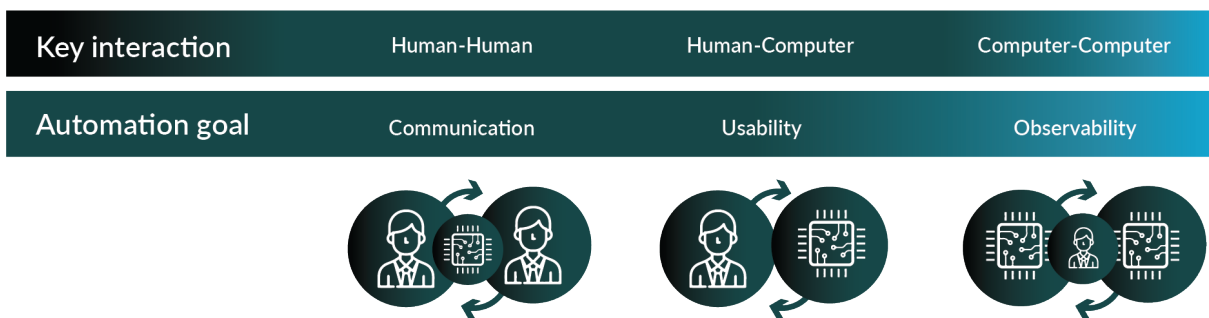
Developer experiences and automation

Once you know where the business value is and how you want to measure it, you can start to think of [the experiences](#) you want to [enable for your software](#)

[developers](#) (and everyone who works with them). Your strategy for defining, designing, and delivering good developer experiences [sets the stage](#) for providing value to customers, employees, and other application audiences.

The key to [good developer experiences](#) is to automate what provides the biggest value to them (and the business) at each stage. Making software is inherently a creative process and [a social activity](#), so [psychology](#) and culture matter. It's not as simple as automating everything and letting the bots do it all. Some approaches to consider:

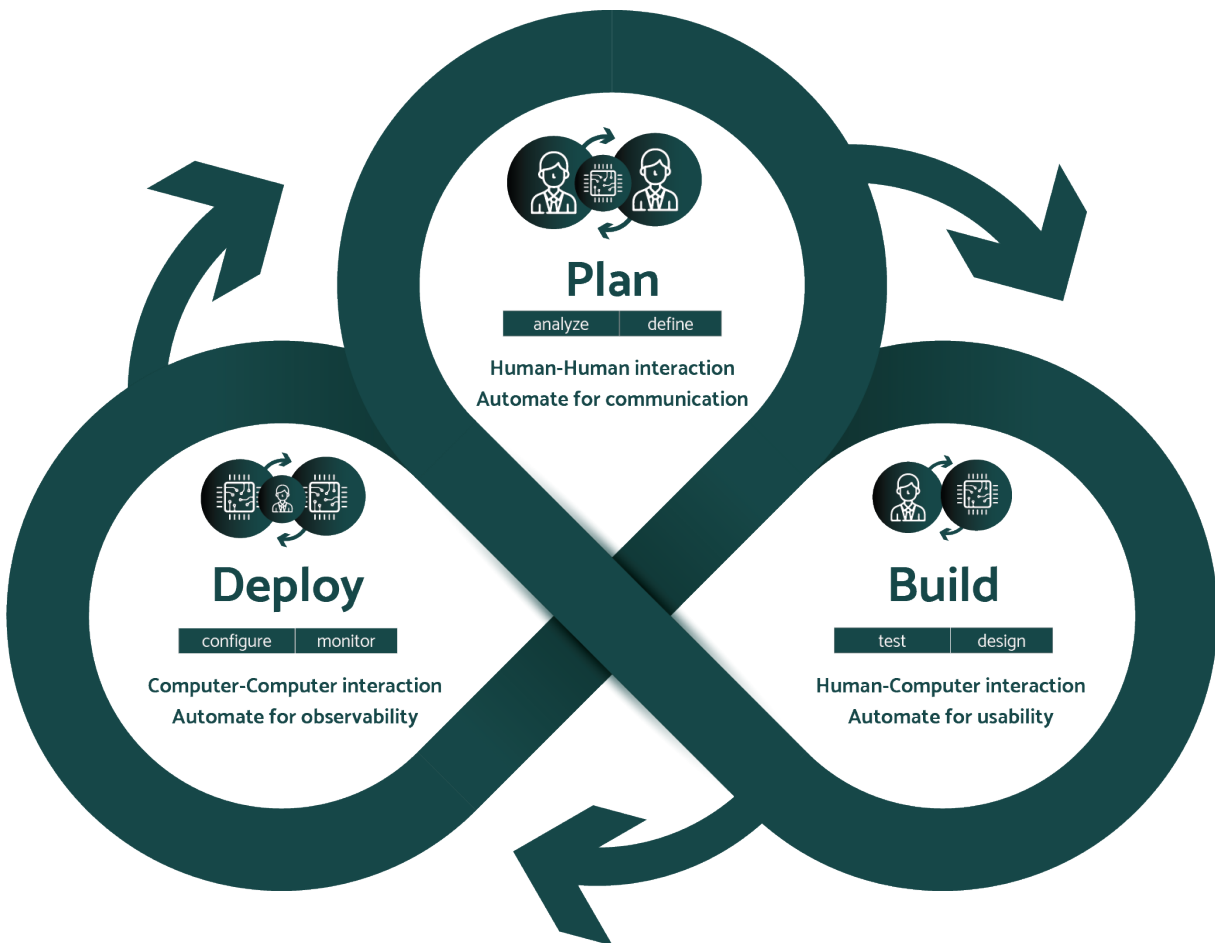
- In the planning stage, improving developer experiences should focus on communication with stakeholders. The key is people working together and analyzing requirements to define what the product, service, or set of features is intended to be.
- In the building stage, efficient and effective designing, coding, and testing of the application provide the highest value. Organizations should focus on providing usable tools that let developers get into a creative flow.
- With deployment, automation shifts focus to the operate portion of the product life cycle. Organizations should focus tooling on monitoring and change management (such as Infrastructure as Code). Also, automate data collection for [business, user, and system performance](#). This drives feedback loops on user experience and business value.



Key types of interaction determine the type of automation to implement

Summary

Your value stream map and management plan, combined with your developer experience strategy, will serve as trusted guides for your automation journey. Your value stream map tells you which parts of your processes are most inefficient, and which have the highest value. Your understanding of when to automate for communication, usability, and observability guides the type of tools you choose and how to integrate them into workflows. In the end, you've built a way for your organization to enable software developer [superhumans](#) to maximize value and innovate.



The role of humans in an automated software development loop