Prototypes – A Misunderstood Opportunity

Iterative design of digital government services

Sebastian Meier, Potsdam University of Applied Sciences, sebastian.meier@fh-potsdam.de
Benjamin Seibel, CityLAB Berlin, benjamin.seibel@ts.berlin
Ingo Hinterding, CityLAB Berlin, ingo.hinterding@ts.berlin

Authors

Sebastian Meier

Sebastian Meier is professor for the design of knowledge transfer at Potsdam University of Applied Sciences Potsdam (FHP). He has a background in design and informatics. At the Interaction Design Lab (IDL) he runs research projects on data-driven innovation and human-computer interactions, for example in the context of urban development, participation, climate change and open data. Before joining FHP, he was overseeing research and development at CityLAB Berlin. He is co-founder of the CityVis.io initiative and chair of the research commission of the German cartographic association (DGfK).

Benjamin Seibel

Benjamin Seibel is the Director of CityLAB, Berlin's first public innovation lab. He is an expert and public speaker with many years of experience in e-government, the development of digital prototypes and the strategic and technical management of open source and open data-related projects involving public administration, academia, and civil society. He published a PhD thesis on the history of digital governance and worked as a journalist and curator in New York, Nicosia, Rotterdam, and Frankfurt/Main.

Ingo Hinterding

Ingo Hinterding is the Product Lead at CityLAB Berlin. He studied visual communication with a focus on interface design at the University of Applied Sciences Aachen and has many years of experience in product management, design and software development both as a freelancer and as a founder of various startups. His focus is on the agile development of pragmatic software solutions for complex problems.

Preface

This workshop contribution is based on the author’s experiences collaborating with the state of Berlin, Hamburg, and Brandenburg, as well as the city of Potsdam and federal organizations like the federal environmental agency (Umweltbundesamt) for more than a decade on building digital services. We will particularly draw from two recent examples out of the CityLAB Berlin: 1) Badegewässer Berlin (Bathing waters Berlin) and 2) Open Traffic Count. This contribution is a strongly auto-ethnographic work, building on the author’s joint experiences working with government organizations.

Introduction

Throughout industries, agile and iterative approaches have become the standard for modern design and development processes, particularly in the domain of digital services. In recent years many government organizations are also keen on adopting such approaches. As it is often difficult to quickly change organizational structures, which had decades to manifest themselves, new organizations outside those structures are created. Examples for such organizations are 18F, a technology and design consultancy unit inside the US government, the Government Digital Services (GDS) in the UK, the DigitalService[s] in Germany, a state-owned design and development agency supporting Germany's federal government, or smaller organizations like the CityLAB Berlin funded by the city-state of Berlin. While these units are meant to work agile and quickly iterate ideas towards new digital services in line with requirements and needs of the citizens, they still need to collaborate with the existing structures inside the administration, to get new products and services into production and to the citizens. These interfaces between those two organizational cultures proof to be challenging in many aspects. In our contribution, we want to focus on one challenge and opportunity, that we have observed in many of our projects: "prototypes" as a misunderstood opportunity for developing digital services.

Preface: Prototypes vs. Specification Guidelines

To better describe the potentials of prototypes within the development of new digital services, we want to contextualize their use and contrast it by traditional government software development. The latter still define the playing field, which new actors, like the organizations described above, need to navigate. The traditional process of building a new software for a government organization can be broken down into the following steps (simplified): 1. The need for a new software needs to be established. As acquiring funding is difficult, there needs to be a "solid" reason to spend money on software development, e.g., a software is technologically outdated or is missing a functionality required by a change in a government process, law, etc. 2. A team is formed to describe the specifications of the software. In most cases the team solely consists of higher-level government employees working on or rather overseeing the procedure in question (Fachverfahren). 3. This team will create a usually very detailed specification guideline (Lastenheft / Leistungsbeschreibung), which describes how the new software should work and look like. The government tech-provider will provide information on requirements regarding the existing infrastructure the new service needs to run on. 4. Those specifications are used to set up a public tender / procurement process. 5. The winning company will then start fulfilling the specification guidelines and strictly stick to it, as they will be held accountable by that document. Problems identified along the way can become an obstacle, as acquiring, and providing additional funding can become a challenge. Especially in large software development projects, the overall process can take years, until a service is available to the users. This highly linear process, driven by organizational hierarchies stands in a strong contrast to modern agile processes (see fig. 1).

When we compare this process to how modern agile development teams work, we can identify several differences. It starts with the beginning of the project and the definition of the target objective. While the traditional process starts by describing in detail what the solution should look like, a modern development process starts by identifying and describing what the actual problem is. Technologies are always a means to an end, and not the focus of the development process. To achieve this, the team developing the project is multi-disciplinary and tries to acquire requirements and needs from a variety of stakeholders, who will be using or profiting from the software after its been build. Instead of describing the solution in the beginning, an agile and iterative process works towards the solution. The concept behind the latter is, to identify

potential challenges earlier on in the process and, thereby, more quickly improve the system throughout the process. In no way is the modern approach faster or cheaper. The big difference is that the product at the end, is better in line with the actual needs and requirements and ideally problems are already identified and solved during the development process. To work towards an implemented digital service, prototypes play an important part. Along the development process a variety of prototypes are used to test, refine, and improve the latest stage of the project.
The CityLAB Berlin, as an exemplary organization in civic tech development, uses this type of agile process in two scenarios: 1) along the overall development process, the prototypes are getting more sophisticated, until they become the actual deliverable (Use case I – Badegewässer). 2) Instead of a release candidate, the prototype is the deliverable, which serves as a concept for the planning / development of the actual solution (by another organization / company). In the latter scenario, the prototype could serve as the specification guideline (Use case II – Open Traffic Count).

From Specifications to Prototypes

Specification guidelines as a basis for software development do not only consume a lot of resources for their production, but they also only provide a static inflexible corset for designers and developers to work in, difficult to change and adapt as the project progresses. We believe that there needs to be a shift away from specification guidelines as a starting point for public software development projects. Through the examples below, we are trying to argue that the *prototypes themselves developed throughout the iterative development process, could potentially service as more holistic specifications*. For example, paper prototypes and wireframes can serve as ways refining requirements and needs, as well as interaction flows throughout the application. Design prototypes can be used to generate design guidelines, which are not abstract, but are informed by exemplary screens or rather applications, which can be further refined along the development process. High Fidelity prototypes, which many civic tech organisations, like Berlin's CityLAB produce, can be used to gain insights into the technical requirements from the necessary data sets to the technical infrastructure to potential data protection pitfalls. By being able to interact with the system in a realistic manner, a variety of stakeholders can give realistic feedback. This approach also comes with it challenges. For example, a high-fidelity prototypes is, for non-technology-experts difficult to differentiate from an actual release candidate software (see fig. 2). As the design and the interactions are very close to a potential release candidate. Explaining that the underlying software and infrastructure are in parts experimental and not meant to be used as a stable release is a challenge. We still believe, that deriving and refining specifications from prototypes along the way provides a development process closer in line with the needs of the target audiences, as well as the needs of the designers and developers themselves building the application.

Use Cases

I. Badegewässer Berlin / Bathing waters Berlin
Berlin's State Office for Health and Social Affairs (LAGeSo) is responsible for measuring and communicating the water quality for the public bathing locations along Berlin's rivers and lakes. The workflow of acquiring samples, processing the samples and publishing the results on the LAGeSo's website can take up multiple days. In the research project Flusshygiene (Seis et al. 2019) a consortium of organizations developed a prediction system and a web-based interface for citizens to get daily water quality information. At the beginning of the project many requirements were not clear, neither from the citizen's perspective, nor from a technical infrastructure perspective. Through an iterative development process, we jointly explored user-, data- and infrastructure-requirements. This helped us quickly develop a first high-fidelity

prototype. We kept the prototype running for more than a year, allowing us to gain further insights, e.g., identify outliers and problems (at that point not defined in any specifications of the partner infrastructures). Through many iterations the release candidate was developed. The two organizational cultures collided, as we tried to transfer the finalized application to the government's infrastructure. A software, developed through such a process, outside the government's specialized procedure (Fachverfahren) protocol, cannot easily be integrated into the public technology infrastructure. As it is, for example, missing a clear specification guideline. Which confronts us with the question, of how to derive those specifications from such a prototype at the end of the development process.

II. Open Traffic Count

Open Traffic Count (Mihaljevic et al. 2021) was a research project exploring the potentials of state-of-the-art computer vision and machine learning techniques for real-time traffic counting, as a privacy preserving alternative to traditional techniques, which require video recordings or manual observation. Not only since the introduction of the European GDPR, data protection and privacy has always been a sensitive issue in Germany. Confronting the regulatory data protection office with a concept of installing video cameras in public places is, for good reasons, likely to raise a lot of red flags. Within our research project, we used a prototype (hardware + software) of the proposed technology, to lead the discussion with the regulatory officers, allowing us to demonstrate and explain easily how data flows, how it is processes in a realistic manner. Without having any specification guidelines or even a full data protection concept at the time, this allowed us, based on the prototype to quickly proceed with the project and install it in a trial run. In this case, the prototype was used to generate the information to fill out the necessary forms and lead the discussion with the administration. In contrast to the first use case, this was a success, and could serve as an example of how we can use prototypes to build specifications and translate the insights we can generate through prototypes into the official required documents and procedures.

Vision

The last case highlights the potentials of *prototypes, as discursive artefacts to inform the process of specification guideline productions*. We are not arguing to abandon the written specification of digital services. We are trying to highlight, the way these documents are being produced and at what point in the development or rather conceptualization process they are being generated, might need to be modified, so they can profit from user-centric and agile approaches. Specification guidelines should be informed by validated requirements (user, technology, data protection, etc.). Such requirements cannot be generated in theory on an empty drawing board, those aspects need to be researched, developed, and tested in a realistic manner. The iterative process of development with and through prototypes is one possibility to generate the necessary insights.

As prototyping tools get more advanced and building high fidelity prototypes is getting easier, one could even imagine, that in future development processes, we could generate specifications fully automated through algorithmic analysis of prototypes. Like how, already today, design guidelines can be extracted from design documents or for example documentations of application programmable interfaces (APIs), databases or cloud infrastructures can be automatically generated from the underlying code. Analogous an algorithm could generate a data protection specification from the interactions with a digital system and the underlying database infrastructure from within a prototype. Changing how the two cultures interact with one another and creating a closer intertwined planning and development process, could potentially also increase the chances, that a lot of the (open source) civic tech, developed by various stakeholders (not only the government funded organizations), find their way into production, serving citizens, and not remain as prototypical concepts.

References

Helena Mihaljevic, Christian Jamal Larsen, Sebastian Meier, Wilhelmina Nekoto & Fabian Morón Zirfas (2021) Privacy-centred data-driven innovation in the smart city. Exemplary use case of traffic counting, Urban, Planning and Transport Research, 9:1, 425-448, DOI: 10.1080/21650020.2021.1950044

Seis, Wolfgang; Meier, Sebastian; Osaki, Mikio; Hemmers, Lukas; Sagebiel, Daniel; Hoppe, Sebastian; Köhler, Antje; Gnirß, Regina; Rouault, Pascale; Szewzyk, Regine. 2019. Entwicklung eines Frühwarnsystems für die Berliner Unterhavel, Korrespondenz Wasserwirtschaft (KW), ISSN: 1865-9926, Jg.12, Nr. 9, 2019

Appendix – Figures:

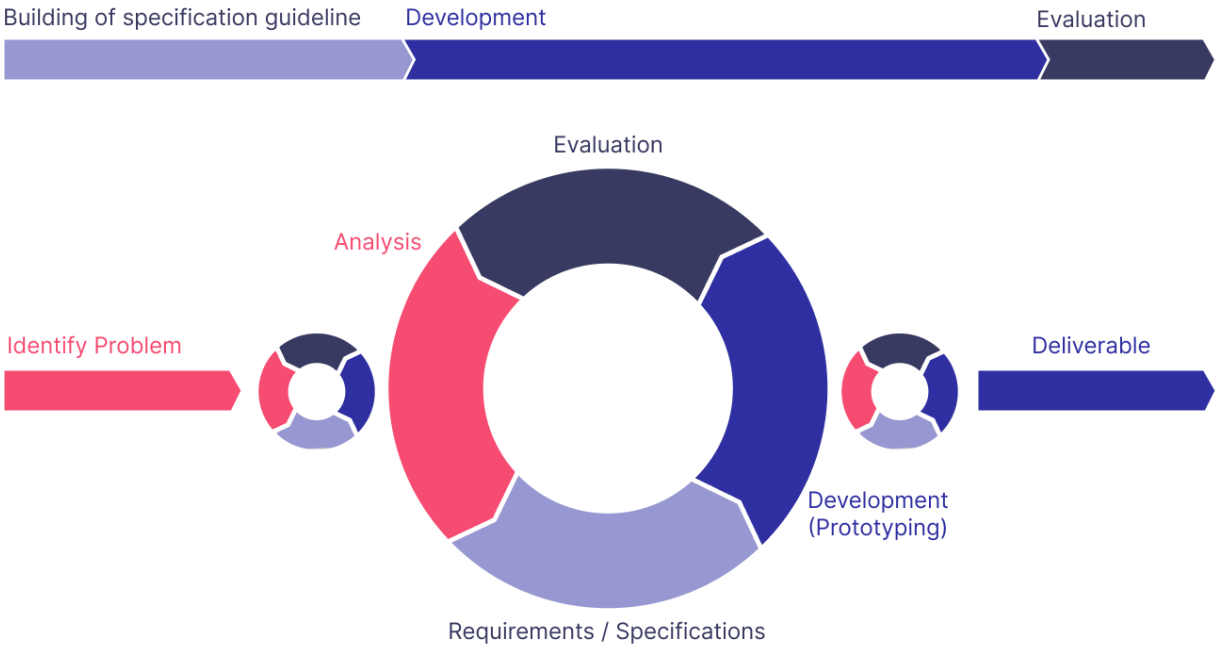

Figure 1: Comparison of a linear (top) and iterative (bottom) development process

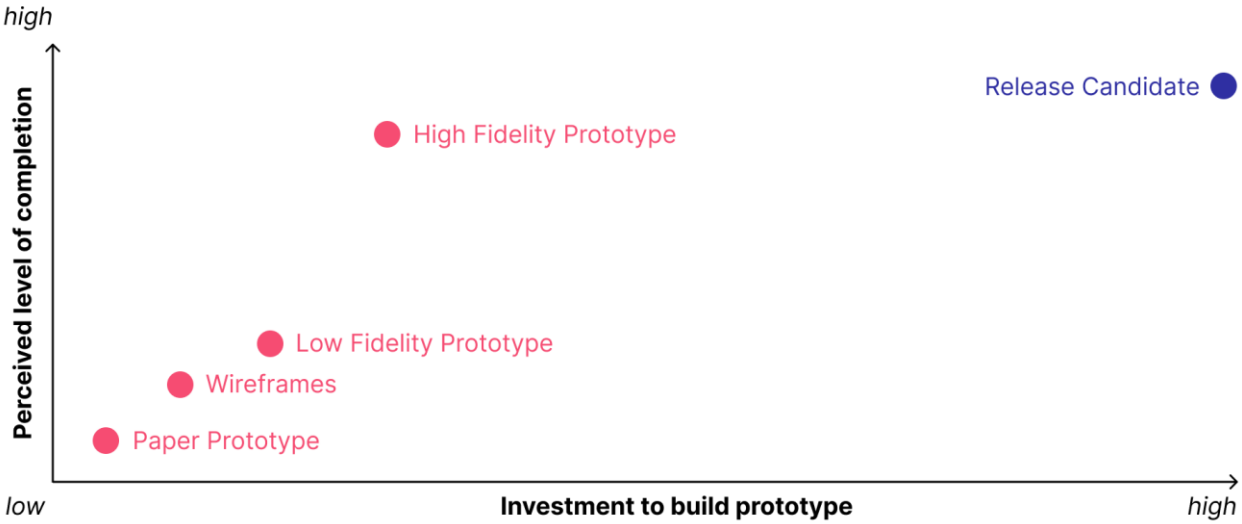

Figure 2: Perceived level of completion vs. Required investment to build a specific prototype/application of a concept.

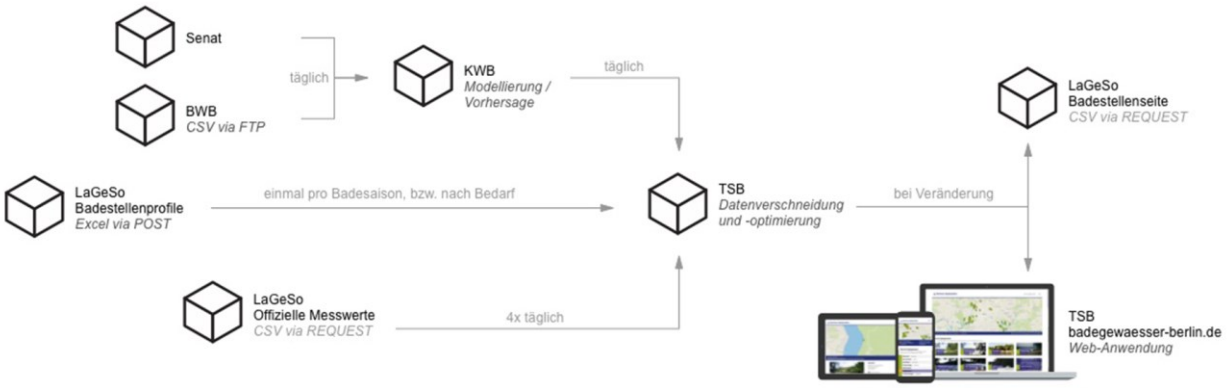

Figure 3: Abstracted data flow schematic for the bathing water quality application (badestellen.berlin.de)

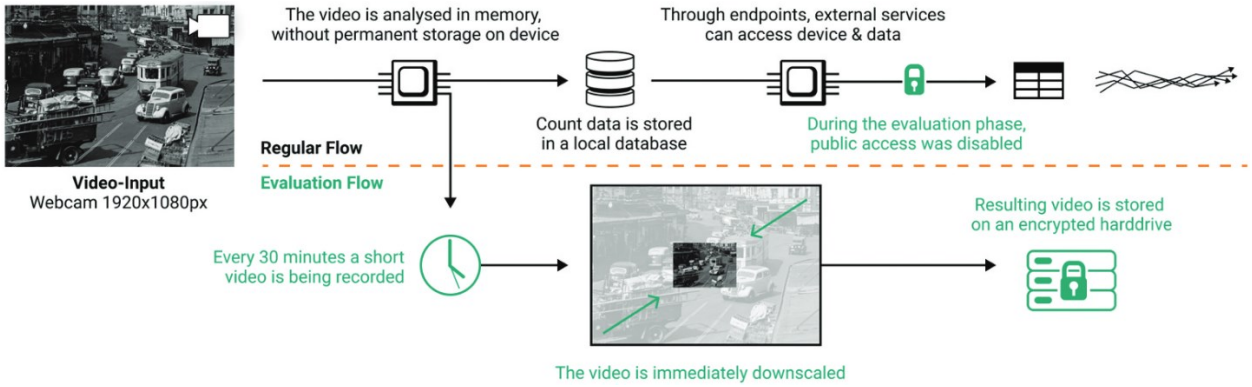

Figure 4: Abstracted data flow schematic for the research conducted in the Open Traffic Count project (figure taken from H. Mihaljevic et al. 2021)