

Translating User Experience to Requirements

Ji Gao, Glen Anderson, Brian W. Bramlett, Ryan Palmer, and Delbert Marsh II

5200 NE Elam Young Pkwy,
Hillsboro, OR 97124, USA

{jade.gao,glen.j.anderson,brian.bramlett,ryan.palmer,
delbert.k.marsh.ii}@intel.com

Abstract. In this paper we introduce the Usage-to-Platform Requirements (U2PR) process in the context of a platform product lifecycle at Intel that involves challenges beyond the development of an individual product. We describe the types of research necessary to inform the process, the methods we have evolved in developing use cases and usage requirements, and the practice used to communicate the requirements to the right audiences. The U2PR process provides a practical approach to fill the gap between descriptions of marketing opportunities and actionable, technology capability definitions and experience-quality requirements. It demonstrates how one can apply a general user-centered design process to strategic planning and design.

Keywords: platform, user experience, usage model, use case, requirement.

1 Introduction

A platform, as understood within Intel, is a set of products that one or more companies deliver to the market as an integrated solution. One of the principal aspects of Intel's transformation from a microprocessor company into a platform company is the building of a comprehensive customer focus to complement technology excellence. Delivery of platform value depends on standardization, and an ecosystem of products and services that support and enable the delivery. Because open, evolving platforms are not limited to a single product type, or even a product line, they are especially challenging to design. Adding to the challenge, the same platform may be delivered through a range of business models in multiple countries, may need to accommodate multiple types of users and uses, and may be required to change rapidly to accommodate changes in technology. In order to manage these challenges, the business, usage, and technology perspectives have to be integrated as a coherent system, referred to as Intel's "Three-Circle Model" [1]. The Usage to Platform Requirements (U2PR) plays an important role in linking all three aspects together.

Prior to the implementation of U2PR process, platform planners and technology architects had difficulty interpreting the usage models identified by Intel marketing teams. In particular, the usage model descriptions did not include the detail necessary to generate actionable engineering requirements. A high-level usage model such as "consuming media anywhere and anytime within the home" had a good deal of thought and research behind it, and the planners also had anticipated what

technologies would likely be employed to provide such an experience; however, handing off the concept to architects with only this level of detail had several problems. The U2PR process was created to specify the usage details and translate it to the usage requirements that could help technology architects design the platform appropriately to deliver the intended user experience. It found its first implementation in the work in 2004 aimed at the Enterprise and Home platforms planned for 2007 and now has been adopted for multiple platforms across Intel.

1.1 U2PR in the Platform Product Lifecycle

The focus of usage-oriented roles evolves over the course of the platform product lifecycle as knowledge and decisions shape the platform. For most of platform or product companies, there are four major phases in the Platform Product Lifecycle (PPLC) characterized as discover, design, develop, and deploy, as shown in Figure 1.

The discover phase is focused on identifying opportunities and establishing strategy. Emerging usage trend-analysis and synthesis from research identify usage models and architectures. The design phase concentrates on transforming strategy into actionable and effective plans. The bulk of U2PR efforts occur during this phase, as key use cases determine specific platform requirements and feasibility. In the develop phase, engineering consumes platform requirements while generating product specifications and implementation details. Development teams must also determine whether the products under development meet U2PR targets for experience quality and usability. The deploy phase deals with getting products built, delivered, and supported. Experience design can encompass the entire “usage lifecycle,” ranging from the initial search for a product to the out-of-box experience to ongoing maintenance. U2PR is a key component of delivering on the promises intended during early planning, and of realizing the value of the platform. The following sections will introduce the U2PR process and the scope of work in greater detail in the context of PPLC, as demonstrated in Figure 2.



Fig. 1. Platform Product Lifecycle (PPLC) phases

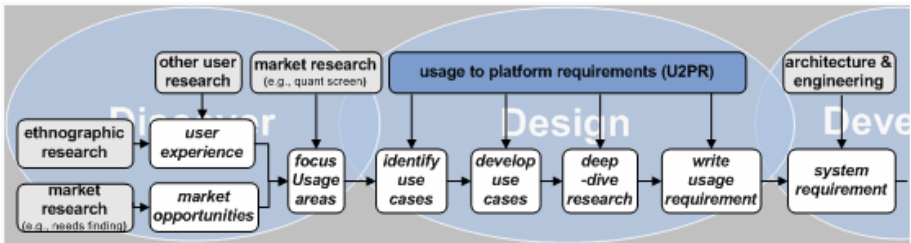


Fig. 2. Role of U2PR in the Platform Product Lifecycle (PPLC)

2 Before U2PR – Identifying Usage Models

Before the U2PR process begins, a process called usage modeling gives Intel strategists a tool for depicting how a particular technology will be perceived and adopted within the context of people's lives. Usage models enable a diverse group of technologists, strategists, and researchers within the company to begin to understand a product's impact on people before the product is actually built. Such information is gathered by market research, ethnographic studies, and review of existing products, etc. The development of usage models ensures a balance between business, technology, and user goals. Once a usage model has been identified as an opportunity for Intel, it is up to U2PR professionals to dig deeper by depicting customer experience, ease-of-use, system interactions, and value propositions, as shown in Figure 2. In preparation for U2PR work, U2PR professionals must have an adequate understanding of the potential business impact and current technology solutions and a thorough understanding of user needs.

3 U2PR Process

The core steps within U2PR are identifying the use cases, developing the use cases, conducting deep-dive research, and writing usage requirements, as shown in Figure 2.

3.1 Identifying Use Cases

As stated earlier, a usage model describes the interaction between a user and a system and demonstrates the benefit of the system to the user within a specific context. A use case defines a more specific sequence of interactions between one or more actors and the system. More than one use case is usually needed to fully define a usage model.

In the example usage model cited earlier, "consuming media anywhere and anytime within the home," the U2PR professional would develop several key use cases to specify this usage model. For example - user selects and streams music from a PC to a device in the home; user adds music to a favorite playlist from any device on the home network; user manages playlists and streams to any device on the home network. It is important to identify a key set of use cases that cover the user types, device types, and media types that would be critical to consider for this usage model.

For platform companies such as Intel, multiple use cases are typically contained within a usage model, and usage requirements enable one or more use cases. While the platform company itself may not develop everything necessary to implement all of the use cases, it is still important to consider the usage model as a whole. This allows the platform company to identify needs and potential gaps for products provided by other companies in the ecosystem. Products to fulfill those needs and gaps can usually be categorized as hardware, software, services, or standards.

3.2 Developing Use Cases

In developing the details of use cases the main objective is to communicate the intended usage experience to others involved in developing a product. The primary

vehicle for this communication is the use-case document. Use cases can take many different forms, which are intended for various purposes. Table 1 depicts the format and description of the elements that comprise Intel use cases as implemented by the User Centered Design Group.

Table 1. Use case elements

Identification
Use Case Title: Short name for reference purposes—the title should accurately represent its contents to avoid misinterpretation or confusion.
Use Case Tag: Unique identifier. Allows for traceability and common reference.
Summary: Brief description to highlight the main activity.
Context
Actors: People/systems who play a part in the use case. They may include formally defined personas (see discussion below).
Pre-conditions: Any pre-requisites or triggers for the use case.
Post-conditions: The end condition, the outcome of the use case.
Body
Sample Scenario(s): Narrative story, depicting a realistic situation, to demonstrate how the actors interact to satisfy the objective of the use case and the context in which the activity occurs. This is invaluable for communicating the intended user experience to architects and others.
Basic course of Events: Task flow visually representing the flow of information and interaction between the actors
Conceptualization: Screen shot, conceptual mockup, storyboard, or other graphical representation of usages to help the audience visualize the intended user experience.
Simultaneous Usage: Usages that may occur at the same time on the same system.
Alternate Paths: Narrative story depicting situations in which the interaction between the actors has deviated from the normal flow. It is important to capture the cause for deviation, and distinctions and considerations from the normal situation.
Exceptions: Listing of the instances when the use case would NOT be triggered.
Requirements Management
Priority: Rating to indicate the importance this use case holds with respect to meeting the usage model objectives.
Author: Name of the individual who wrote/developed use case for tracking and reference purposes for any questions regarding the specifics of the use case.
Usage Owner/Originator: Name of the individual who conceptualized the usage for tracking/reference purposes for any questions regarding the origins of the use case.

In developing use cases, it may be valuable to take an approach that Cockburn terms “Black Box Use Cases” [2]. With this approach the use case does not address specific technologies but focuses instead on the user and the desired usage experience,

leaving the implementation open for alternatives and design innovation. At other times a technology may be central to defining how a use case unfolds. In these cases, the U2PR professionals may take a pragmatic approach that includes a particular technology, which may involve some limitations to the user experience, as part of the use case.

3.3 Deep-Dive Research

One approach that has been used to inform the U2PR work is referred to as “deep-dive” research. This approach is a combination of human factors and design research methodologies, which are commonly used in contextual inquiry and task analysis. The purpose of deep-dive research is to better understand user needs in the context of identified usage models and therefore to validate use cases and capture important requirements. As Figure 2 indicates, the deep-dive takes place at a later stage of PPLC. The findings from deep-dive research may require that the use cases be modified.

U2PR professionals conduct deep-dive research via a series of face-to-face, semi-structured interviews with individuals who are identified as representative of users that have shown interests in a particular usage. The first objective of the research is to understand the pain points in participants’ current user experience by encouraging users to talk about their daily work, frustrations, and even walking through typical tasks. The second objective is to gather user feedback on proposed technology solutions that enable the usage to determine what features are needed and how the platform should be designed to ensure the delivery of a desired user experience. The interviewer introduces the participants to a solution through conceptualizations (e.g., storyboards) and records their responses to pre-designed questions and elicits ratings of the appeal and relevance of various attributes of the proposed solution.

The essential output of ‘deep-dive’ is detailed information that allows usage models to be developed into a set of usage requirements, as explained in the following section. The deep-dive and the followed brainstorm and ideation activities would be helpful discovering new usages. A wide variety of methodologies may be applied to ideation and are covered in depth in Kelley [3]. Though low-fidelity prototypes and story boards that show example user interfaces may be useful in this type of research, deep-dive research generally precedes product design.

3.4 Writing Good Requirements

The requirements coming out of the ‘deep-dive’ research is referred to as usage requirements, which directly describe the human experience of the system. The technical requirements come from other sources, such as technical specifications and standards. The main tenets of effective usage requirement include complete, correct, clear, concise, consistent, coherent, connected, feasible, prioritized, and verifiable [4]. Going beyond these basic tenets, the most important consideration when constructing usage requirements is to at all times develop them with the users in mind. What are the users’ objectives? What interactions are reasonable to expect of them in the context of the activity that is taking place? Usage requirements must lay the necessary framework for the experience to satisfy the user.

Requirement writers should compose in a language that is easily interpreted by systems architects or software developers. The usage requirements communicate affirmative statements that clearly articulate what should be developed to improve the experience people have with the product. Each requirement also contains a rationale for why it needs to be considered when developing the technology to support the usage. The rationale points to primary and secondary research and data supporting the requirement. As with use cases, the focus is on the fundamental need to be met, not a specific technological implementation. An example of a usage requirement is as follows:

Name: Secondary Recording Status Display

Description: The PC shall present recording status when the PC is in the process of recording a TV show or capturing digital content, even if the main display is turned off.

Rationale: TV recording devices typically show a status light on the bezel to show recording in progress, allowing the user to see status without turning on a connected screen.

Priority: High – essential for competitive offering

Usage requirements can be categorized as functional and non-functional. The example above is a functional requirement, since it describes a particular function the system must perform. A non-functional requirement defines a quality of the system, such as how good the perceived video quality must be. Both types are outputs of the U2PR process.

4 After U2PR – Communicating Usage Requirements

The first key element to ensure the requirements are well consumed is getting requirements to the right people. Usage requirements are communicated both formally and informally to a diverse team of engineers, architects, software developers, planners, and designers involved with product development across the platform. To ensure user-oriented concerns are considered along with technology-centric approaches to platform architecture, user requirements are communicated with the rationale derived from research and human factors expertise. Mapping usage requirements to key technology capabilities allows architects to reference the end-user value that the technology enables. Reliance on more tangible implementations of the user experience is important. Aligned with this, U2PR professionals communicate requirements in the form of compelling storyboards, videos, vignettes, and concept prototypes. Visual conceptualizations also provide a tool for eliciting feedback on future usages that do not have an existing frame of reference. In addition, U2PR professionals use personas [5] in the context of a use case to show how a user's values and experience play a role in the user's interaction with a system. The persona gives product teams an opportunity to take a walk in target users' shoes and experience their goals and behaviors.

The U2PR at Intel ends with a set of usage requirements that feed an architectural analysis process. In some organizations, the same group that does a U2PR-like process may do this work as well, but for the scope of this paper, U2PR ends with the

communication of the usage requirements. However, the user experience torch should be carried through prototype evaluation and post-shipment, through a “user experience quality” measurement program.

5 Conclusion

Companies that develop platforms composed of multiple products face special challenges in filling the gap between marketing plans and platform architecture. This creates key roles for U2PR professionals who can communicate across these product groups and organizations, and it requires the corporate to invest resources to enable the creation of a coherent platform. Another challenge is getting U2PR professionals in the correct mindset to define a platform, instead of only one product that will present the user interface. It requires the close work with ecosystem partners to ensure success in the final platform. Those challenges go beyond those inherent in individual product development. The U2PR process fills the gap using a combination of research, human factors methodologies, use case and requirements processes, and visual conceptualization.

Acknowledgements. The authors thank Paul Sorenson for recognizing the need for a U2PR process at Intel, for establishing the program, and for ongoing leadership and guidance. We thank Erik Simmons for helping to define the process and thank Herman D’hooge for fostering a corporate culture that recognizes the need to bring the user perspective into technology development.

References

1. Simmons, E.: The Usage Model: Describing Product Usage during Design and Development, *IEEE Software*, vol. 23, no. 3, May/June, (2006) pp. 34–41
2. Cockburn, A.: *Writing Effective Use Cases*, Addison-Wesley, Boston, MA, ISBN 0-201-70225-8, (2001)
3. Kelley, T.: *The Art of Innovation*, Doubleday, New York, NY, (2001)
4. Hooks, I. and Farry, K.: *Customer-Centered Products, Creating Successful Products through Smart Requirements Management*, Amacom, ISBN 0-8144-0568-1, (2001).
5. Cooper, A.: *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*, SAMS Publishing, Indianapolis, IN, (1999)