User-Centered Evaluation of the Responsiveness of Applications

Gerd Waloszek and Ulrich Kreichgauer

SAP User Experience, SAP AG, Germany

Abstract. This paper presents an approach to evaluating the responsiveness of software applications from a 'user's perspective, which has been developed and applied at SAP AG, Germany, a leading manufacturer of business software. The approach is based on human time ranges and compares actual with tolerable response times, measured using standardized application scenarios.

Keywords: System responsiveness, perceived performance, human time ranges

1   Introduction

Making software applications highly responsive is a usability and quality goal of utmost importance for software companies: It determines how efficient the users of a software application can be. Usually responsiveness is approached from a technical perspective. At SAP AG, a leading manufacturer of business software, dedicated technical teams measure the responsiveness of applications in clearly defined test environments, using step-by-step scenarios to compare different software versions by evaluating the effects of technical fine-tuning to improve responsiveness. However, these measurements tell us little about how users experience an application's responsiveness. Therefore SAP's User Experience team initiated the "Perceived Performance (PeP)" project to devise a user-centered method and apply it to the scenario-based measurements of the technical performance teams.

2   User-Related Criteria for Evaluating Responsiveness

2.1 The Challenge

The technical performance teams at SAP evaluate the responsiveness of applications by monitoring a number of parameters, one of these being the overall response time for user-initiated user interface (UI) events. The teams use a one-second threshold as a criterion for whether an application achieves SAP's performance goals. From a user-centered perspective, however, the one-second rule does not reflect expectations or behaviors. Some actions should take less than a second, others may take longer. The challenge is to develop an evaluation method that provides better insight into the actual user experience, and to identify areas needing improvement.

2.2 The Initial Concept: Human Time Ranges

The concept of human time ranges that refer to the psychological dimensions perception, operations, and cognition (thinking, attention, motivation) looked promising as a starting point for developing such a methodology.

Allen Newell's time scales of human action seem to be the original source, even though these were published after related work. In several papers from the end of the 1980s to the beginning of the 1990s, Card, Robertson, and Mackinlay applied the time ranges to human-machine interaction. A number of authors, including Jakob Nielsen (1993), Alan Cooper (2007), and, most notably, Jeff Johnson (2007), adopted the original table and adapted it to their needs. In their most basic and cited form, the time ranges are defined as follows:

- 0.1 sec.: Perception – cause-and-effect, animation > direct manipulation tasks
- 1 sec.: Operation – focused man-machine dialog > simple tasks
- 10 sec.: Cognition – focus on task lost > complex or compound tasks

Shneiderman and Plaisant (2004) mention an additional category of "common tasks" of about three seconds. This category is useful because it also marks two effects that "waiting" has on users. After three seconds (according to some authors, even after one or two seconds), users start to feel that the system is slow and loose their task focus. Nonetheless, they can maintain a degree of focus until up to 10 seconds. The authors also introduce a further relevant time: After waiting 15 seconds, users become annoyed. The PeP team integrated both categories into its time ranges.

2.3 PeP Application of Time Ranges

Table 1. PeP adaptation of human time ranges table, including variations in ()

Time Range	Human Aspect	Application / User Interface (UI): Acceptable Response	User: Response When Feedback Does Not Meet Time Range
0.1 sec. (0.0-0.2)	Perception	Acknowledges user input	Perception of smooth animations and cause-and-effect relationship breaks down
1.0 sec. (0.2-2.0)	Dialog, action	Presents result of simple task	Engaged user-system dialog breaks down
3 sec. (2.0-5.0)	Cognition, attention, motivation	Presents result of common task	User has time to think – the system is perceived as slow, the user's focus starts to wander, and the user may turn to other tasks
10 sec. (5.0-15)		Presents result of complex tasks	User looses focus on task and may turn to other tasks
>15 sec.		Presents result of very complex task	User becomes annoyed – the system is detrimental to productivity and motivation

The basic idea behind the PeP method is to classify observed response times according to time ranges, and thus the psychological effects on users of waiting. This requires extending and connecting the time ranges from 0 to beyond 15 seconds, with-

out leaving any gaps. The PeP team adopted Shneiderman's and Plaisant's (2004) values for the variation of the time ranges wherever possible, but a few decisions could not be backed up with data from the literature. We therefore initially set fairly conservative upper limits for the time ranges: See the first column in table 1.

2.4 PeP Assignment of UI Events to Time Ranges

By assigning UI events to the time ranges, you get a clearer picture of the response time a user with some experience would expect, and thus tolerate. The above-mentioned authors provide some suggestions for assignments, but for practical use in its evaluations, the PeP team created the following list:

- *Level 0: 0.1 (0-0.2) seconds – Perceptual Level:* Feedback after UI input involving direct manipulation/hand-eye coordination, such as mouse click, mouse/pointer movement, key press, button press, menu open/close.
- *Level 1: 1 (0.2-2) seconds – Dialog Level:* Finishing simple tasks, that is, most user-requested operations and ordinary user commands, finishing unrequested and system-initiated operations, opening a window (navigation) or dialog box, closing a window, completing a simple search.
- *Level 2: 3 (2-5) seconds – Cognitive Level:* Finishing common tasks, such as logging in to a system.
- *Level 3: 10 (5-15) seconds, Level 4: >15 seconds – Cognitive Level:* Completing complex tasks, that is, one task or one step of a multi-step task, completing one step in a wizard, completing a complex search or calculation.

3   The PeP Methodology

The PeP methodology is based on three steps:

1. *Preparation*: We break the use scenarios into task steps, or technically, UI events. These we categorize according to what response time would be tolerable for users. This (preliminary) assignment is based on the complexity of interactions, that is, the workload for the computer that experienced users would expect.

2. *Measurement*: We time the UI events and assign them to the time ranges. This assignment is based on the events' actual duration, and thus on the users' perception, not their expectations.

3. *Evaluation*: This data leads to a frequency matrix of tolerable versus observed time ranges (see table 2), which can be interpreted from a user's perspective.

Since for users' perceptions and reactions, the time ranges have distinct implications (directness, appropriateness, slowness, waning or lost focus, annoyance), the PeP evaluation matrix demonstrates a more detailed evaluation of how users perceive the performance of a software application than an evaluation that is based solely on one fixed time limit. This type of evaluation is particularly valuable if an application is considerably slower than expected or exhibits wide response-time variations.

In the (fictional) example, 30.1% fulfillment rate for simple tasks has a strong negative impact on user satisfaction. Generally, we would argue that the fulfillment

rate should be highest for simple tasks, that is, tasks that need to be accomplished at a fast pace, and can be lowest for complex tasks. In our regular work, we measure many standardized scenarios. Our typical goal is to speed up processes so that observed UI events are at least as fast as the category they have been assigned to (diagonal in matrix); however, goals may be lower or more ambitious.

Table 2. Example of a PeP evaluation matrix (fictional data)

Tolerable Range	Observed Range (Number of Times Measured)				Total	Fulfillment Rate (%)
Type of Interaction	0.2-2.0 s	2.0-5.0 s	5.0-15 s	> 15 s		
Simple Tasks (0.2-2.0 s.)	22	26	20	5	73	30.1
Common Tasks (2.0-5.0 s.)	3	13	9	9	34	47.1
Complex Tasks (5.0-15.0 s.)	0	1	2	1	4	75.0
Overall	25	40	31	15	111	36.9

4   Future Research Directions

Further validation is needed of the assumptions on which the PeP evaluations are grounded. Currently, UI events are assigned to time ranges on a preliminary basis; a systematic investigation is required of actual UI events and their assignment to time ranges. Secondly, the transition points between the time ranges are based on data from the literature and on heuristic assumptions. These should be verified in systematic experiments involving users who rate the timeliness of selected UI events.

Another idea is to investigate the value of developing performance-oriented guidelines. This would entail defining high-level rules on top of UI guidelines for specific applications. Another idea is to measure the time costs of UI controls and suggest alternative designs to reduce screen rendering times.

5   References

1. Card, S., Robertson, G., and Mackinlay, J. (1991). The information visualizer: An information workspace. *Proc. ACM CHI'91 Conf.*, 181-188.
2. Alan Cooper, Robert M. Reimann & Dave Cronin (2007). *About Face 3.0: The Essentials of Design*. John Wiley & Sons, 220-221.
3. Jeff Johnson (2007). *GUI Bloopers 2.0: Common User Interface Design Don'ts and Do's*. Morgan Kaufmann Publishers (Chapters 1 & 7).
4. Allen Newell (1994). *Unified Theories of Cognition*. Harvard University Press.
5. Ben Shneiderman and Cathérine Plaisant (2004). *Designing the User Interface (4th Edition)*. Pearson Addison-Wesley (Chapter 11, p. 454ff).
6. Jakob Nielsen (1993). *Usability Engineering*. San Diego, CA: Academic Press. (Chapter 5)
7. Robertson, G., Card, S., and Mackinlay, J. (1989). The Cognitive Co-Processor Architecture for Interactive User Interfaces. *Proceedings of the ACM Conference on User Interface Software and Technology (UIST '89)*, pp 10-18. New York: ACM Press.
8. Robertson, G., Card, S., and Mackinlay, J. (1993). Information Visualization Using 3D Interactive Animation. *Communications of the ACM*, 36(4), 56-71.