

Creative Software Engineering

What is it?

Software engineering is traditionally seen as very structured and methodical. However, it often involves creative steps, such as requirements analysis, architecture engineering and GUI design.


Why do we need it?

The author hopes to re-provide technological experiences in physical contexts, broadening access to facilities such as social websites. Methods such as TAPT (below) help us understand user experiences.

Existing Techniques

Software engineers use the following complementary tools to understand how user groups may perceive and use technologies. This helps focus on users, find assumptions, and communicate clearly.

Personas are descriptions of people representative of certain user groups. We might define name, age, gender, job, ethnicity, family, socioeconomic status... Personas are based on real data.



Persona: Tom

Tom is a teacher of French in a secondary school. He has recently taken on the responsibility of head of department. He is very interested in innovative ways of teaching languages and has visited some schools that use blended learning.

Scenarios show situations in which technology is used. They are stories, with a setting, characters and plot. They help practitioners reflect on designs and find assumptions.

Jess is a manager who receives a weekly automated report on expenditure. When this week's report comes in, she wants to skip straight to the summary, so she selects this option from the menu. The summary is displayed, and Jess identifies an area of interest, which she would like to investigate. She then...

Architectural patterns are reusable solutions to common software design problems. They are abstract templates to be customised and applied to the specific problem at hand, leaving a creative jump between picking up the pattern and the finished solution.

Name: Model-view-controller
Problem: Applications can become cumbersome and complex; splitting them into separate layers can lead to better implementation.
Solution: A three-part model of the following structure...

Teasing Apart, Piecing Together (TAPT)

TAPT is a new process that helps software engineers better understand experiences, allowing the re-design of these in new contexts.

Stage one, teasing apart:

- 1) Briefly describe the functionality
- 2) Describe the surface elements, related to design
- 3) Describe experiential effects:
physical, emotional, intellectual, social
- 4) Review the effects, identify those which seem key
- 5) Use this data to describe the **abstract** experience in one **neutral** sentence.

Teasing apart <u>social status updates</u>				
Experience (1)	Surface elements (2)	Experienced effects (3 & 4)		Distilled experience (5)
		Literal (3.1)	Abstract (3.2)	
Make a comment viewable by others who can annotate it with their own responses	status friends phrase response chain identification brief simple informative humorous sharing	Broadcast info Pushed out Shared Info. pointer	Amused Sympathetic In touch Connection Involvement Feel Heard Recent Prioritised	People share updates to stay in touch with each other
Remember to note key effects!				

Stage two, piecing together:

- 1) Brainstorm on how the distilled experience might manifest in the new context
- 2) Build a scenario
- 3) Check the reconstruction: did you capture everything?
Are there unintended key effects? Adjust accordingly.

Piecing together <u>social status updates</u>	
Distilled experience	People share updates to stay in touch with each other
Context	Handheld Devices (Individual)
Description	A system of updating their status with voice snippets or templated images (emotional and event-based) that other residents can view/listen to and post/record their own responses, supplemented by public screens carrying the feeds. Indexed by photos of individuals.