Rapid Software Architecture Exploration

Michael Keeling
IBM
@michaelkeeling

Lozziwurm @ Carnegie Museum of Art
Amazing experiences do not happen by accident… they are intentionally designed.

Designing amazing experiences is not easy.
Agenda

- Theory
  - The Science of Design
  - Architecture-Centric Design Thinking
- Practice – Workshop!
- Reflection and Discussion
A VERY BRIEF INTRODUCTION TO THE SCIENCE OF DESIGN

Design as an Optimization Problem

Global Maxim

Local Maxim
Goal: Find the Best Solution to the Problem

Knowledge about the Problem

Knowledge about the Solution

Theoretically Best Solution

A Problem: Bounded Rationality

Boundary created by Time, Money, Knowledge, ...

Knowledge about the Problem

Knowledge about the Solution
A Problem: Bounded Rationality

Knowledge about the Problem

Best solution you can actually achieve.

Knowledge about the Solution

More

Less

Rapid Software Architecture Exploration

Cheat bounded rationality with a fast and effective design strategy.
The “Software Design Space”

Knowledge about the Problem

More

Less

Knowledge about the Solution

More

The “Software Design Space”

Cost for knowledge is higher.

Will you get a solution you like?

Slow and Ineffective Design Strategy

Knowledge about the Problem

More

Less

Knowledge about the Solution

More
What makes for a fast and effective design strategy?
The “Software Design Space”

More

Knowledge about the Problem

Less

Knowledge about the Solution

More

More

Less

Less

Big Design Up Front (Waterfall)

More

Knowledge about the Problem

Less

Knowledge about the Solution

“Mountain Summit”
Known Solutions (Platforms)

Routine Problems
Evolutionary Thinking (Agile)

Knowledge about the Problem

More

Less

Knowledge about the Solution

More

Less

Potentially Releasable Software = Best Design Today
ARCHITECTURE CENTRIC DESIGN THINKING

The Problem with “Process”

Many design processes are built on poor assumptions…

• Series of sequential steps
• Problem, solution considered disjoint
• Novel design is objective
• Stops at design (ignores construction)
Design Thinking

A way of thinking.

A set of *modes* that can be executed in any order.

Design Modes

- Explore
- Understand
- Evaluate
- Make
Understand

Actively seek information from stakeholders and work to (re)frame the problem.

Explore

Use generative thinking to identify design concepts and engineering approaches.
Realize design concepts by creating them in the real world as a model, prototype, program, or other artifact.

Evaluate
Determine the fitness of design decisions and decide whether to revisit other modes.

Make
Design Modes

- Explore
- Understand
- Evaluate
- Make

RAPID EXPLORATION WORKSHOP
Ground Rules

• No right or wrong answers
  – Use your experience, imagination to fill in missing details

• Watch the clock (I'll help too)
  – I'll give you time limits
  – When it's time, it's time

• Ask me questions if you need help or clarification

• Help each other
  – Listen for my signals

• HAVE FUN! 😊

• Tweet / Share your experiences!
  – #OReillySACon
Form groups of 3-5 people and introduce yourselves.

Context

Your group is now a team that has been hired to build some software…
Challenge

The city of Boston has hired you to architect a mobile application to help people find and pay for parking (for cars).
Parking App – High Level Features

• As a car driver I can…
  – Find available parking places
  – Pay to park
  – Review and pay parking tickets
• As a policeman I can…
  – Issue parking tickets, find stolen cars
• As a city council member I can…
  – Review historical parking data and metrics

A Context Diagram…

Develop, Deploy, Maintain

Developer

Map open spots, See remaining time

Parker User

Verify Parking

License plate, Parking meter number

Reserve parking

Police User

Issue tickets

City Council User

View Reports

System to Be Built

Payment Processors

Make

Parking Spots Services

Send Payment

Paid / Expired Status

View Reports

Payment Processors

Make
Stakeholder Map

- Explore
- Evaluate
- Understand
- Make
Stakeholder Map

Visualize the relationships, hierarchies, and interactions between all the people who have an interest in the system to be built.

Directions: Add and annotate stakeholders collaboratively until time runs out or the map seems complete (for now).

Guidelines and hints:
- Simple icons represent individual people, label specific role
- Represent categories of people using groups of icons
- Speech bubbles represent thoughts, feelings
- Arrows connect people
- Label lines to describe relationships
Stakeholder Map

Description  A network diagram of the people involved with or impacted by a given system or system design

Time Needed  30 – 45 minutes

Benefits  • Identify more than the usual stakeholders  
          • Document, guide plans for research  
          • Keep the team focused on people rather than technologies

Participants  As many potential stakeholders as available – team, customer, etc.

Practice: Stakeholder Map  10 Minutes

Visualize the relationships, hierarchies, and interactions between all the people who have an interest in the system to be built.

Directions:
Add and annotate stakeholders collaboratively until time runs out or the map seems complete (for now).

Guidelines and hints:
• Simple icons to represent individual people, label specific role  
• Don’t represent categories of people as a single icon  
• Speech bubbles represent thoughts, feelings  
• Arrows connect people  
• Label lines to describe relationships
Failure Modes

- Explore
- Understand
- Evaluate
- Make
Failure Modes

**Description** Consider what failure might look like to shed light on important quality attributes.

**Time Needed** 15 – 45 minutes

**Benefits** • Sometimes easier to articulate failure than success
• Alternative way to look at quality attribute scenarios
• Can jump start other brainstorming activities

**Participants** As many potential stakeholders as available – team, customers, QA, IT, etc.
Quality Attribute

Benchmarks that describe a system’s intended behavior within the environment in which it was built.

Requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.


Quality Attribute Scenario

**Raw Scenario:** Briefly describes the essence in part or full of a scenario.

Environment

Source

User, component

Specific function or action

Stimulus

Artifact

A system component

Response

Observable outcome

Response Measure: [specific, measurable]

Based on work by Rebecca Wirfs-Brock, Joseph Yoder
Practice: Failure Modes 10 Minutes

Identify what failure for the system could look like as a means of understanding success.

Directions:
Brainstorm failure modes for various quality attributes. Use this insight to then create a desirable raw quality attribute scenario.

Guidelines and hints:
• Focus on one quality attribute (*-ility) at a time.
• Raw scenarios are OK – consider: source, stimulus, artifact, response, response measure, environment
• Be as specific as possible.
• Ask extreme questions, e.g. “What if the system were down for 24 hours?”, “What about 1 million users?”

Round Robin Design
Round Robin Design

Explore
Understand
Evaluate
Make
Round Robin Design

**Description** Quickly generate and vet many architecture design ideas through a quick succession of fast peer reviews.

**Time Needed** 60 – 90 minutes

**Benefits**
- Foster creativity by constraining design
- Create opportunities for unplanned combinations.
- Encourage group ownership of the design
- Build consensus among disparate ideas.

**Participants** Team
(other stakeholders can help validate)

---

Practice: Round Robin Design 12 Minutes

Quickly build a collection of ideas for architecting the system, then converge ideas and start to build consensus.

**Directions:**
1. Sketch a single view of the architecture (cartoons OK)
2. Everyone sketches a design (~5 minutes)
3. Pass your design to the person on the left.
4. Critique and annotate the design (~3 minutes)
5. Return papers to original author
6. Briefly discuss insights

**Guidelines and hints:**
- Everyone sketches
- No right or wrong answers
- Use different color ink for each critique
Trade papers!

Practice: Round Robin Design 3 Minutes

Quickly build a collection of ideas for architecting the system, then converge ideas and start to build consensus.

Directions:
1. Sketch a single view of the architecture (cartoons OK)
2. Everyone sketches a design (~5 minutes)
3. Pass your design to the person on the left.
4. Critique and annotate the design (~3 minutes)
5. Return papers to original author
6. Briefly discuss insights

Guidelines and hints:
- Everyone sketches
- No right or wrong answers
- Use different color ink for each critique
Give papers back and discuss!

Question – Comment – Concern
Question – Comment – Concern

Explore  Understand

Evaluate  Make
Question – Comment – Concern

**Description** Brainstorming technique that helps quickly identify and visualize specific areas in the system that may require further thought.

**Time Needed** 30 – 90 minutes

**Benefits**
- Visualize high risk, unknown, or concerning parts of the system
- Promote knowledge sharing
- Identify and visualize areas in need of research or exploration

**Participants** Team, relevant stakeholders

---

Mobile Parking System (Dynamic Perspective)
Practice: Question – Comment – Concern 8 Minutes

Identify questions, comments, and concerns your team has regarding specific models in the architecture.

Directions:
1. Brainstorm questions, comments, and concerns
2. Place sticky dots on the most relevant area of the sketch
3. Annotate the diagram with the question, comment, or concern
4. Reflect and review findings

Guidelines and hints:
• Use different color sticky dots to visualize uncertainty or concern
• Place comments on top of or next to questions or concerns they directly address.
Show and Tell (Reflection)

Show and Tell  5 minutes

• Show one of your worksheets to a nearby team.
• What is something interesting you notice about their work?
• What was something interesting you learned during this session?
Architecture-Centric Methods

**Understand:** Quality attribute scenarios, personas, empathy map, system properties web, architecture drivers specification, user journey, elevator pitch, user mad lib, …

**Explore:** System journey, design the extremes, define the design concept, yours and mine list, round-robin, estimate the elements, system research, soap boxing, paths not taken, name the styles/patterns, …

**Make:** Create a template, mock-ups, paper prototype, functional prototype, sketches/cartoons, architecturally evident coding style, system metaphor, architecture haiku, context diagram, utility tree, module decomposition, viewpoints and views, must reads list, …

**Evaluate:** Scenario walk-through, dot voting, I like/I wish/what if, feedback capture grid, risk Storming, bull’s eye, Question-Comment-Concern,…
Three Fs of Good Design Methods

Fast
Effective
Fun
Thank you!

Michael Keeling
IBM
@michaelkeeling
http://neverletdown.net