Rapid Software Architecture Exploration

Michael Keeling
IBM
@michaelkeeling

Amazing experiences do not happen by accident… they are intentionally designed.

3

Designing amazing experiences is not easy.

4

Agenda

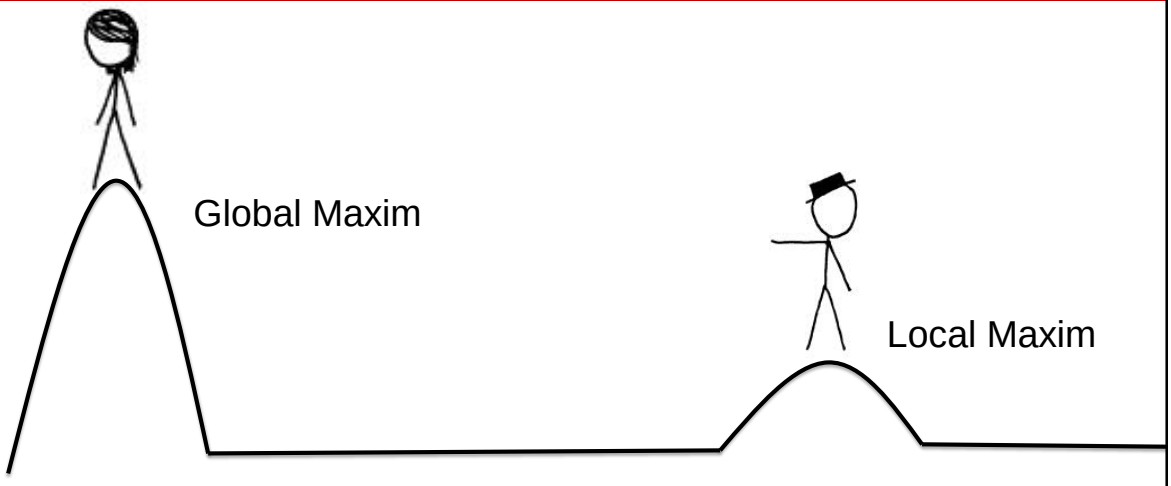
- Theory
  - The Science of Design
  - Architecture-Centric Design Thinking
- Practice – Workshop!
- Reflection and Discussion

6

A VERY BRIEF INTRODUCTION TO THE SCIENCE OF DESIGN

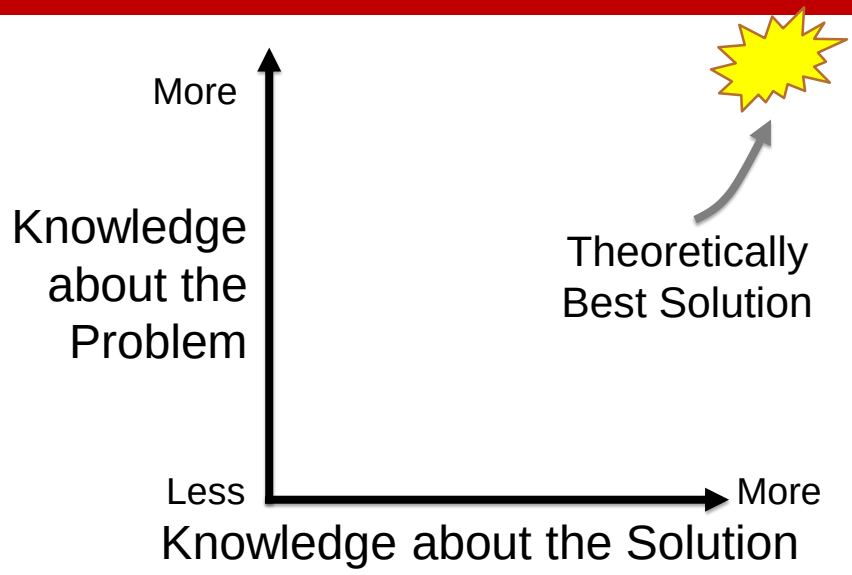
7

Design as an Optimization Problem



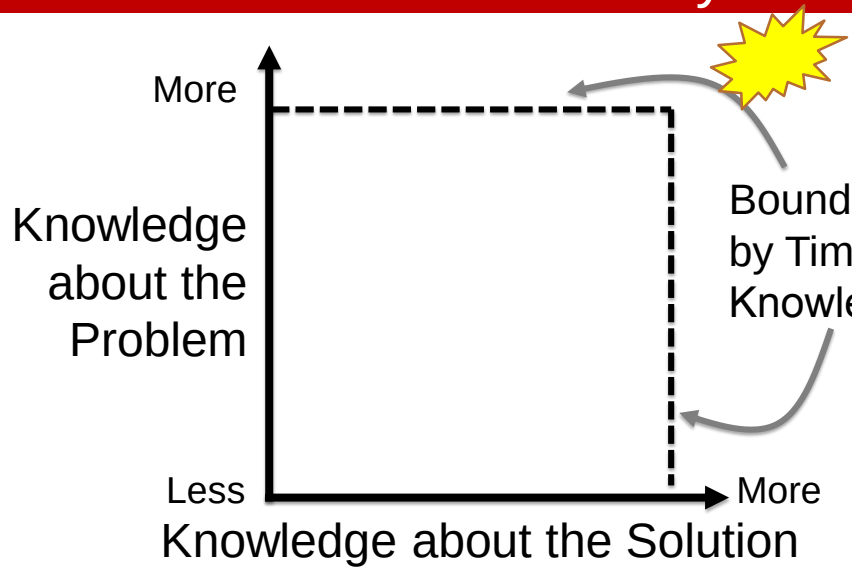
9

Goal: Find the Best Solution to the Problem



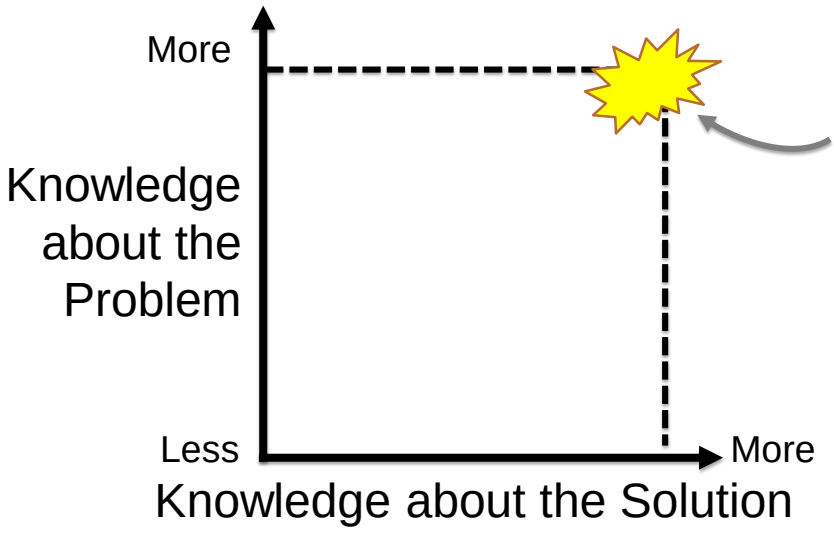
10

A Problem: Bounded Rationality



12

A Problem: Bounded Rationality



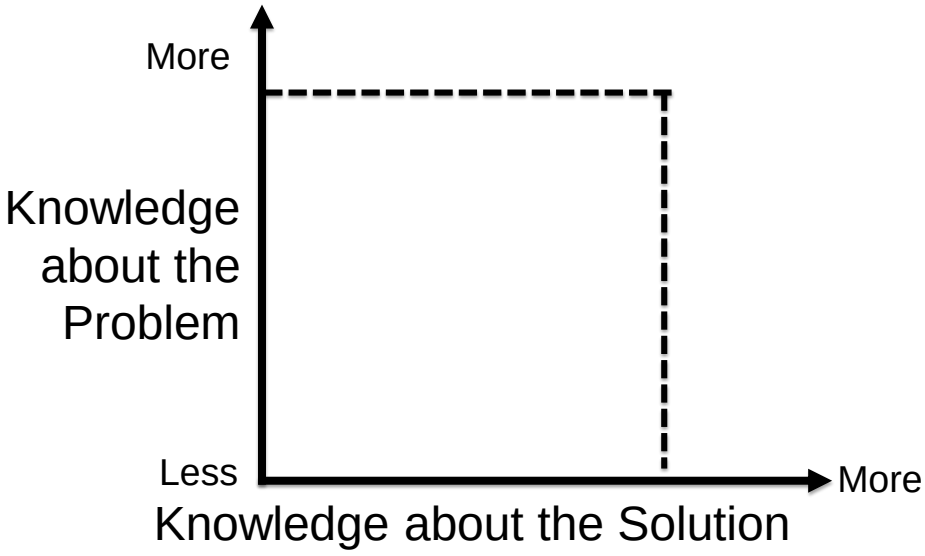
13

Rapid Software Architecture Exploration

Cheat bounded rationality with a ***fast*** and ***effective*** design strategy.

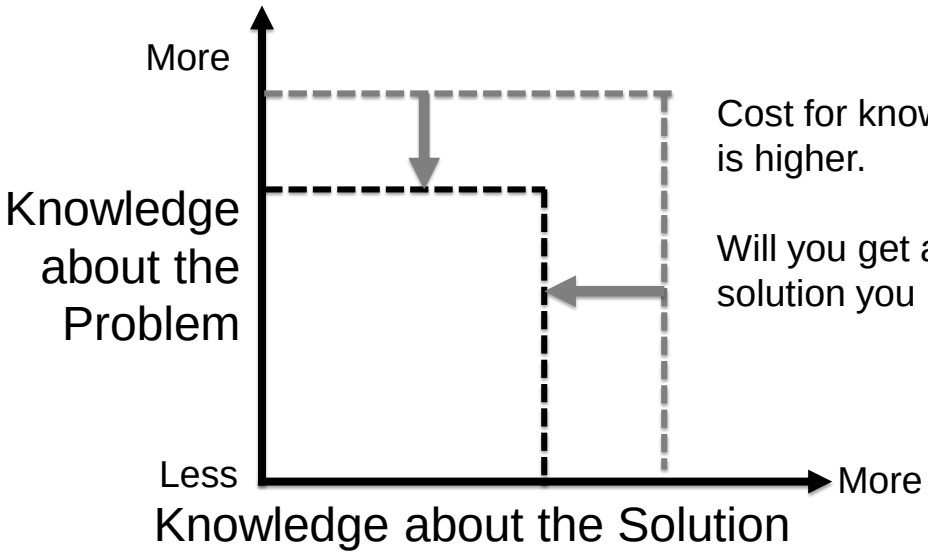
14

The “Software Design Space”



15

Slow and Ineffective Design Strategy

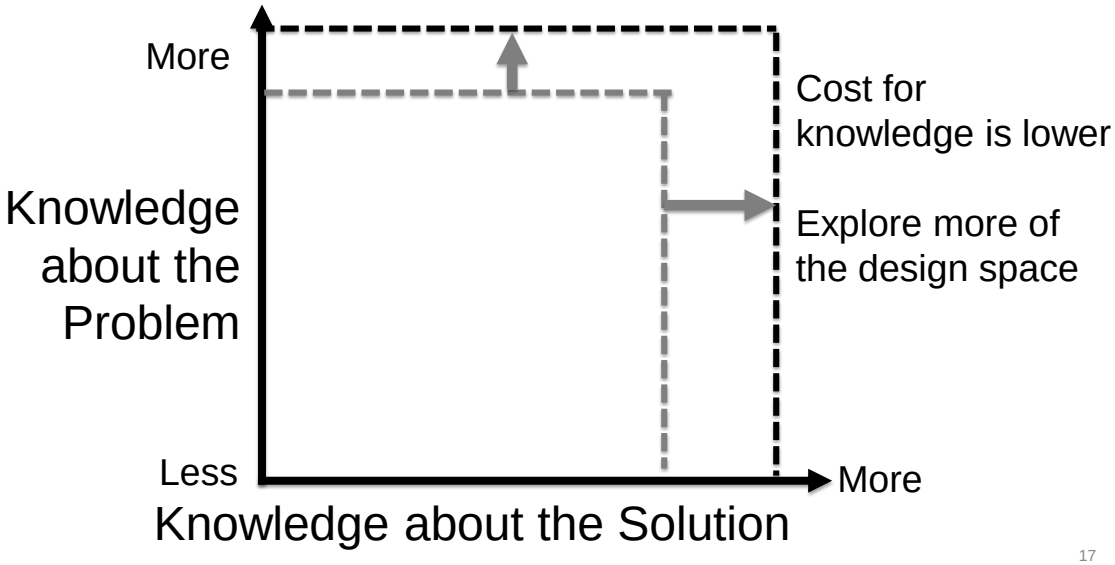


Cost for knowledge is higher.

Will you get a solution you like?

16

Fast and Effective Design Strategy

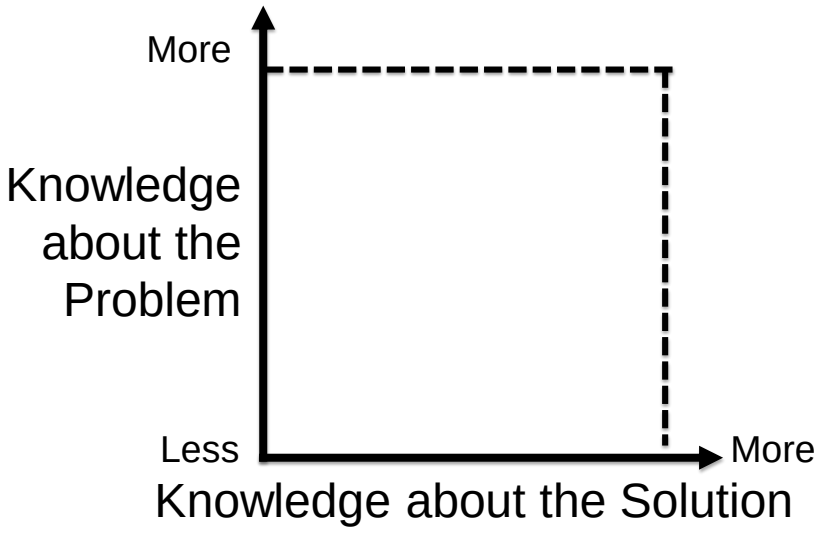


17

What makes for a fast and effective design strategy?

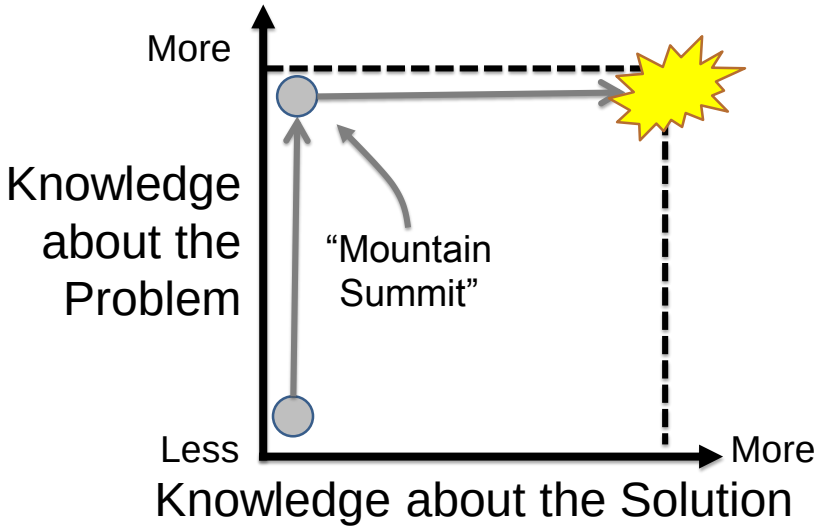
18

The "Software Design Space"



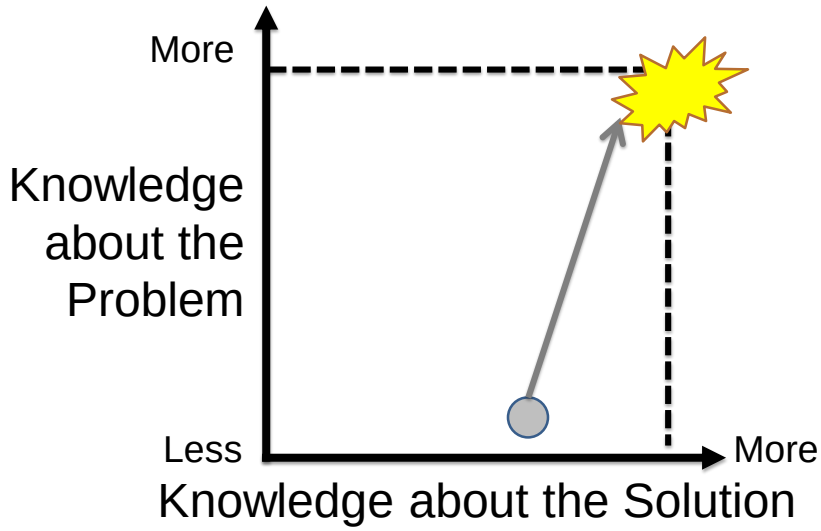
19

Big Design Up Front (Waterfall)



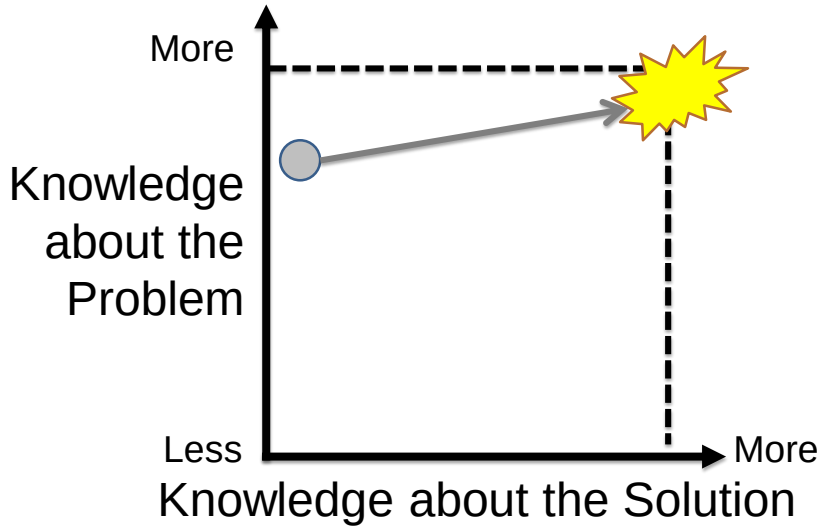
20

Known Solutions (Platforms)



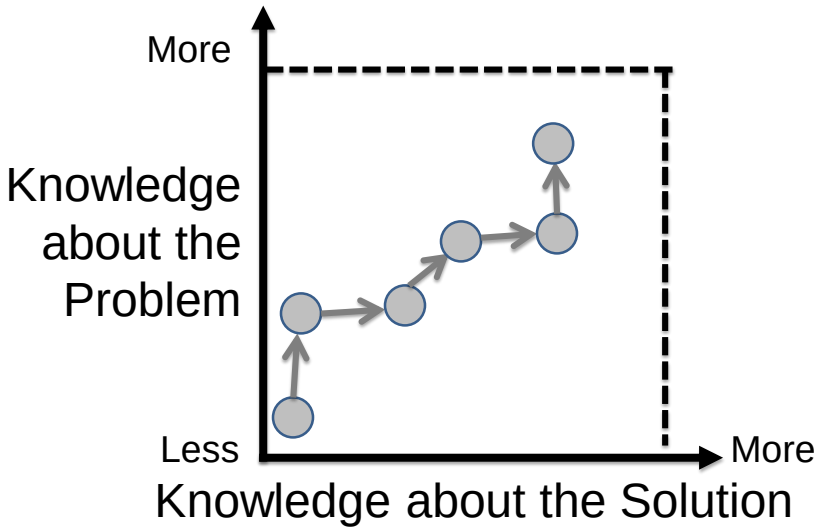
21

Routine Problems



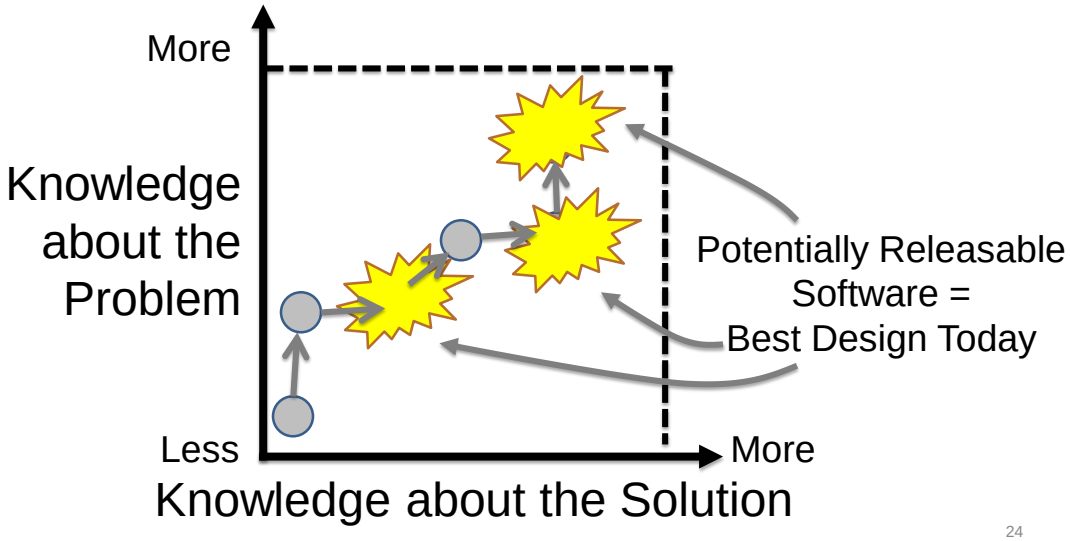
22

Evolutionary Thinking (Agile)



23

Evolutionary Thinking (Agile)



24

ARCHITECTURE CENTRIC DESIGN THINKING

25

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)

26

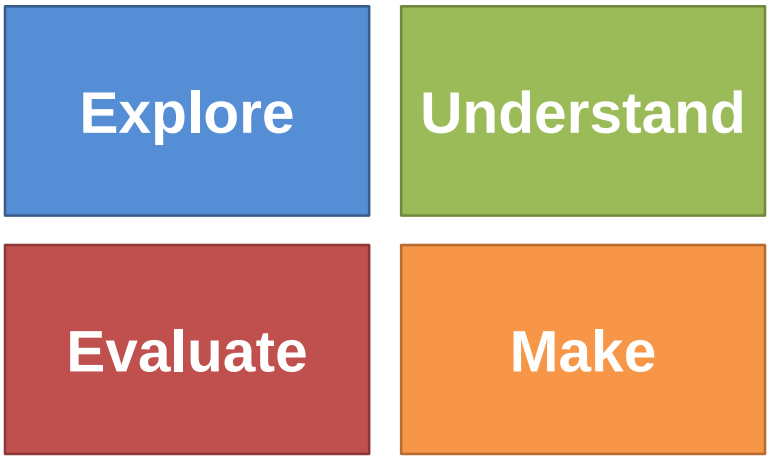
Design Thinking

A way of thinking.

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

27

Design Modes



28

Understand

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

29

Explore

Use generative thinking to identify design concepts and engineering approaches.

30

Realize design concepts by creating them in the real world as a model, prototype, program, or other artifact.

Make

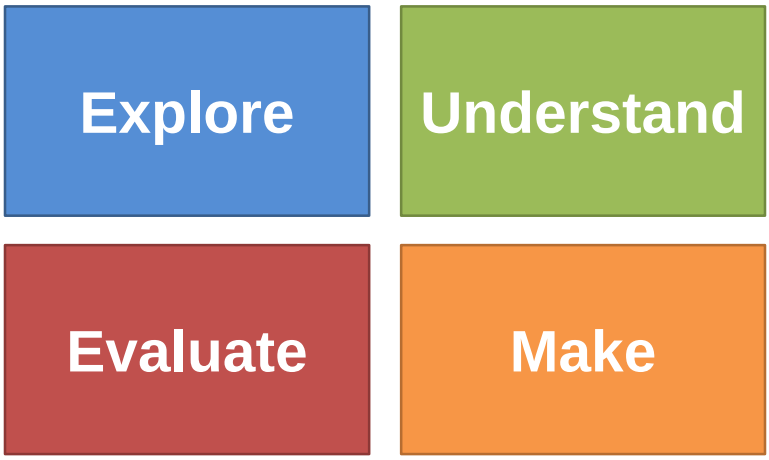
31

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

Evaluate

32

Design Modes



33

RAPID EXPLORATION WORKSHOP

34

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

35

36

Form groups of 3-5 people and introduce yourselves.

37

Context

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

38

Challenge

The city of Boston has hired you to architect a mobile application to help people find and pay for parking (for cars).

39

40

Parking App – High Level Features

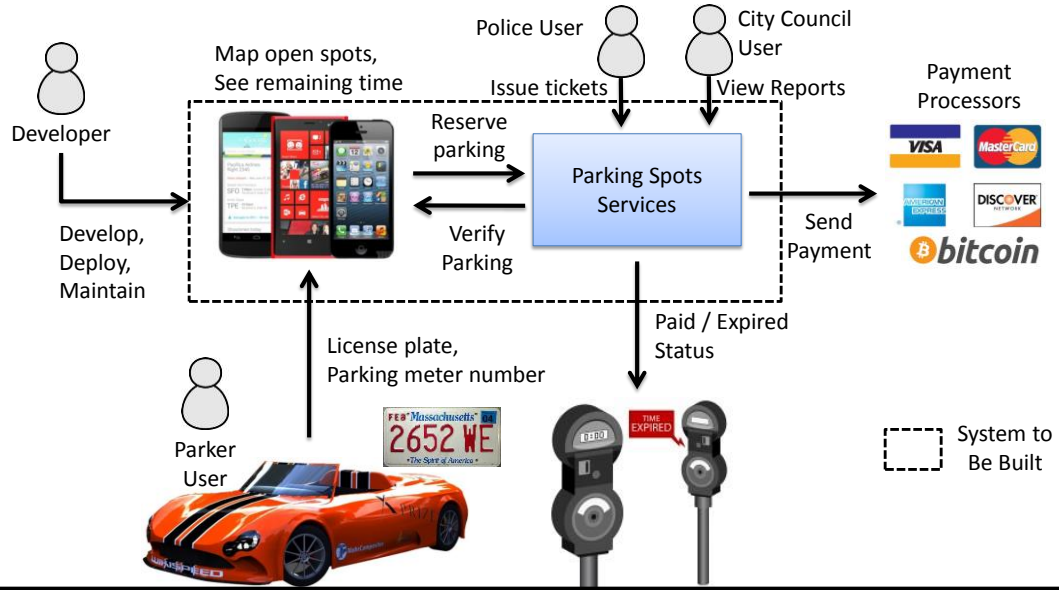
Understand

- 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

41

A Context Diagram…

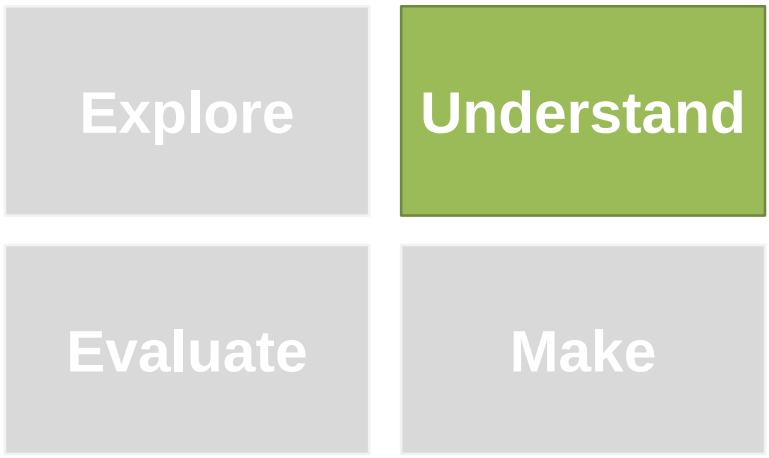
Make



42

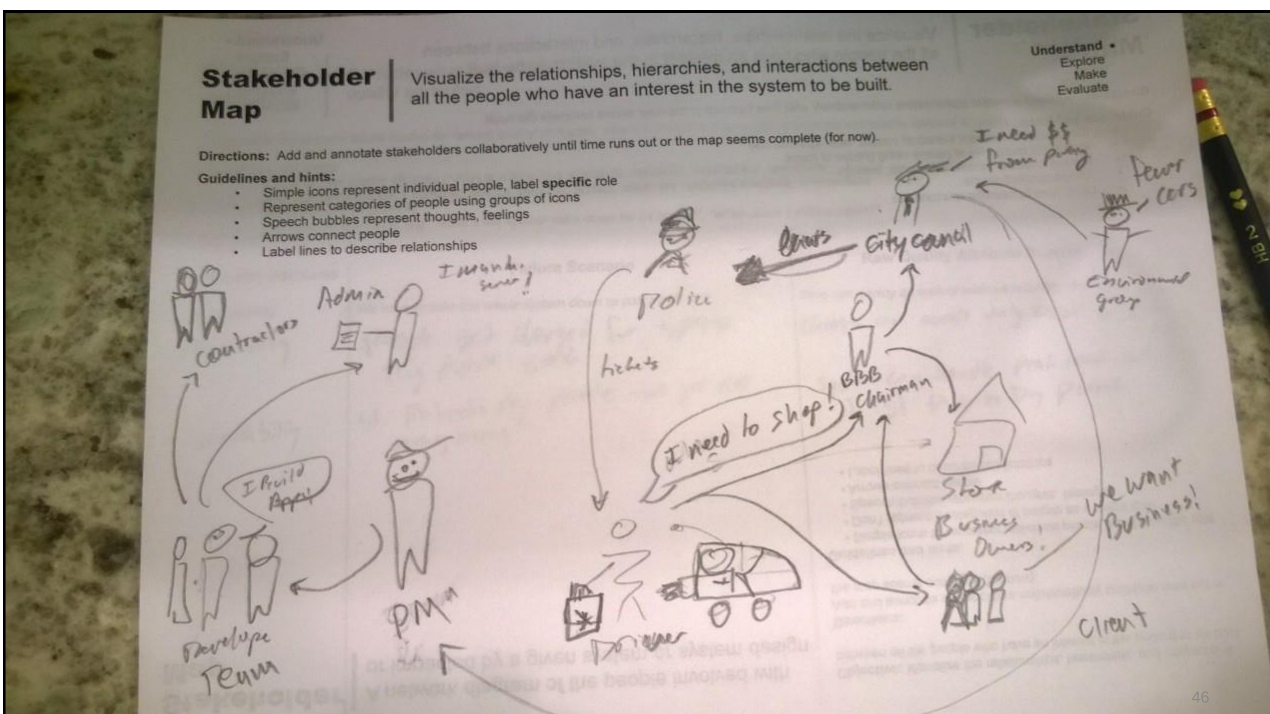
Stakeholder Map

Stakeholder Map




CXO Stakeholder Map
CEO $
CFO $
How Much? What's the ROI?
Need to be QUICK!
I don't even know which products might benefit me!
TARGETED EXISTING CUSTOMER
STRATEGIC ACCOUNT MANAGERS
BUSINESS ANALYST (SUBJECT MATTER EXPERT)
Program Manager
CUSTOMER SERVICE REPS
TARGET CUSTOMERS


Can we afford this?
City
Mayor
do we have enough parking? too much congestion?
Town planner
Administrator
You have 24 hours!
Police chief
Bank/ Payment Provider
Who should I give a ticket to?
Police
Where can I park?
a Driver
where's my car?
I need parking space for my employees & customers
a pedestrian driver.
employer. business owner
I have space I want to make money from
IT


Stakeholder Map
Visualize the relationships, hierarchies, and interactions between all the people who have an interest in the system to be built.
Understand •
Explore
Make
Evaluate
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
Admin
contractors
Police
tickets
City council
I need $$ from parking
Fewer cars
Environmental Group
I need to shop!
BBB Chairman
Store
Business Owners!
We want Business!
Client
I Build Apps
Develope Team
PM
Driver
46

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.

47

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

48

Failure Modes

49

Failure Modes

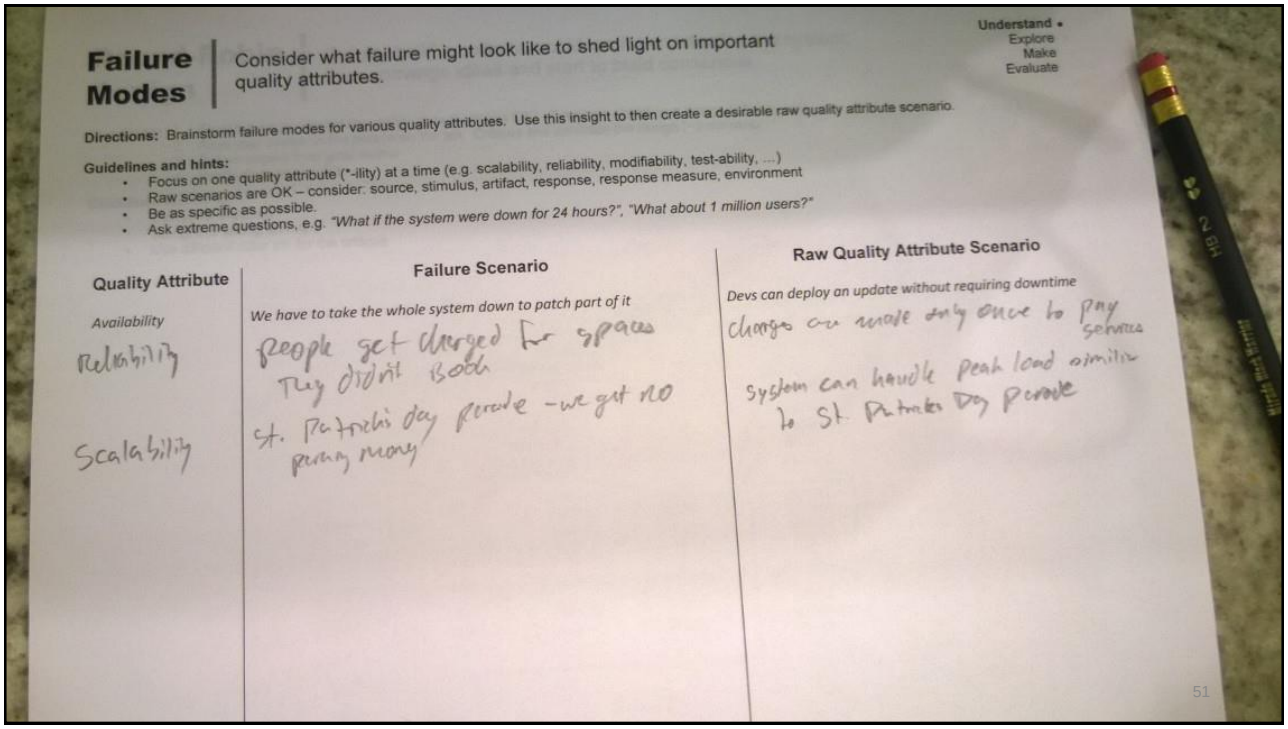
Explore

Understand

Evaluate

Make

50



Failure Modes | Consider what failure might look like to shed light on important quality attributes.

Understand • Explore Make Evaluate

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 (e.g. scalability, reliability, modifiability, test-ability, ...)
- 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?"*

Quality Attribute	Failure Scenario	Raw Quality Attribute Scenario
Availability	*We have to take the whole system down to patch part of it*	*Devs can deploy an update without requiring downtime*
Reliability	people get charged for spaces They didn't book	charges are made only once to pay services
Scalability	St. Patrick's day parade - we get no parking money	System can handle peak load similar to St. Patrick's Day Parade

51

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.

52

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.

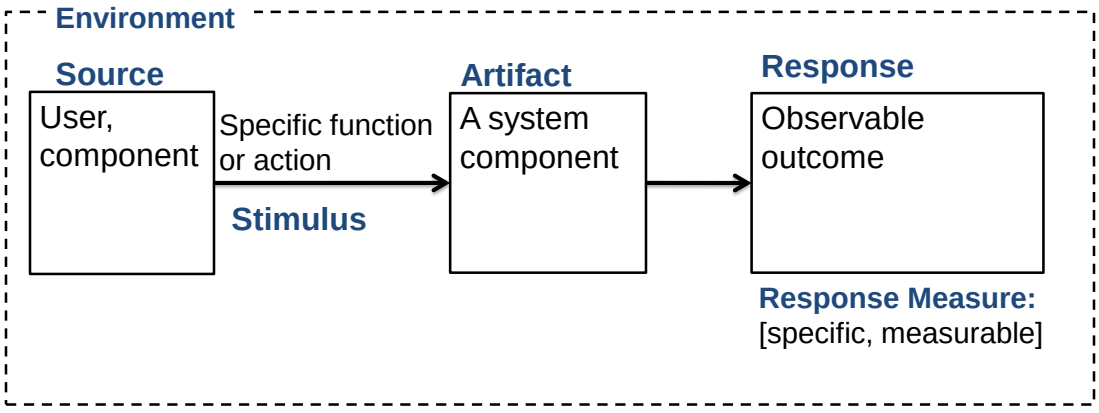
http://en.wikipedia.org/wiki/List_of_system_quality_attributes

http://www.softwarearchitectures.com/go/Discipline/DesigningArchitecture/QualityAttributes/tabid/64/Default.aspx

53

Quality Attribute Scenario

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



Based on work by Rebecca Wirfs-Brock, Joseph Yoder

54

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?"*

55

Round Robin Design

56

Round Robin Design

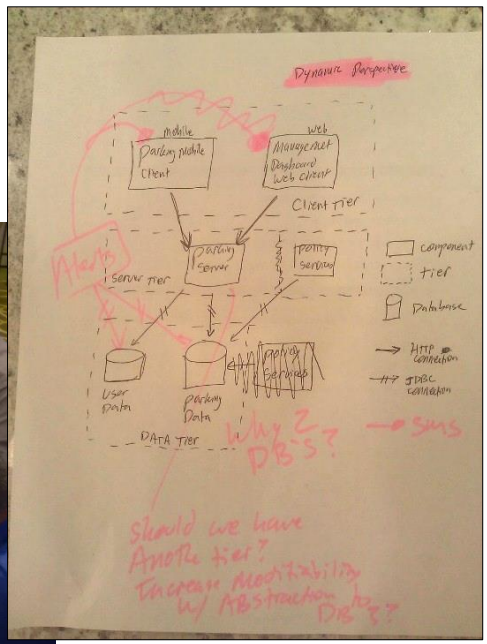
Explore

Understand

Evaluate

Make

57



58

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)

59

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

60

Trade papers!

61

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

62

Give papers back and discuss!

63

Question – Comment – Concern

64

Question – Comment – Concern

Explore	Understand
Evaluate	Make

65

66

Question – Comment – Concern

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

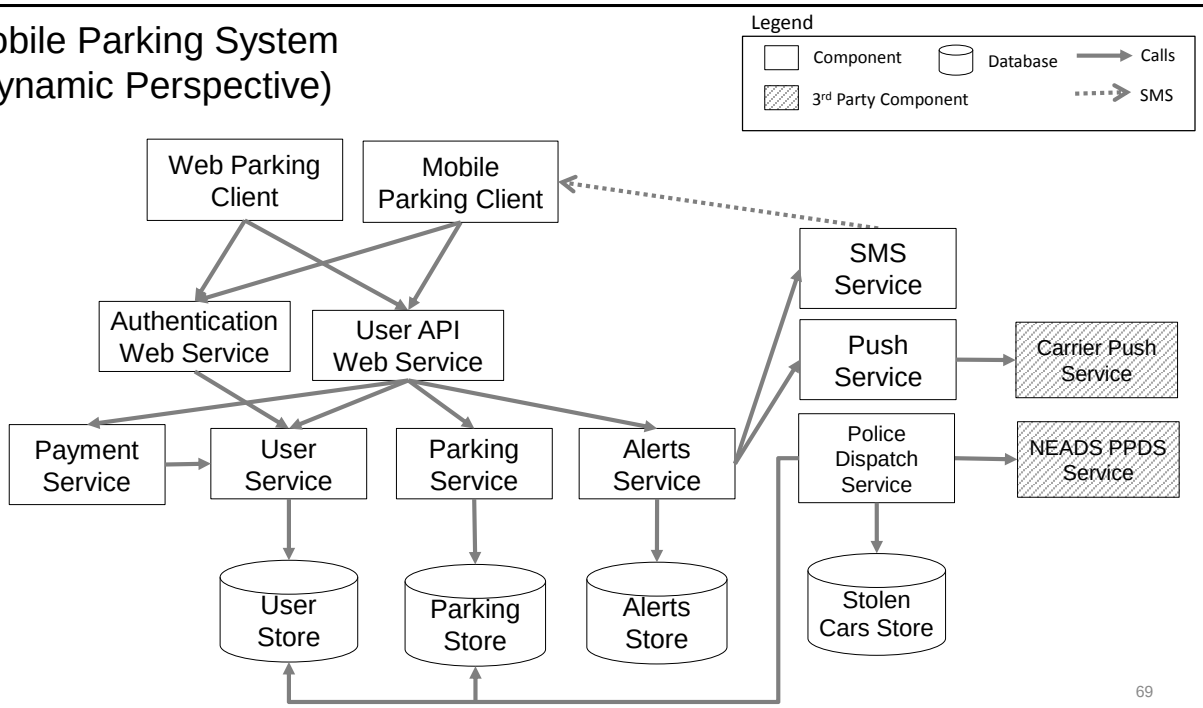
Time Needed 30 – 90 minutes

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

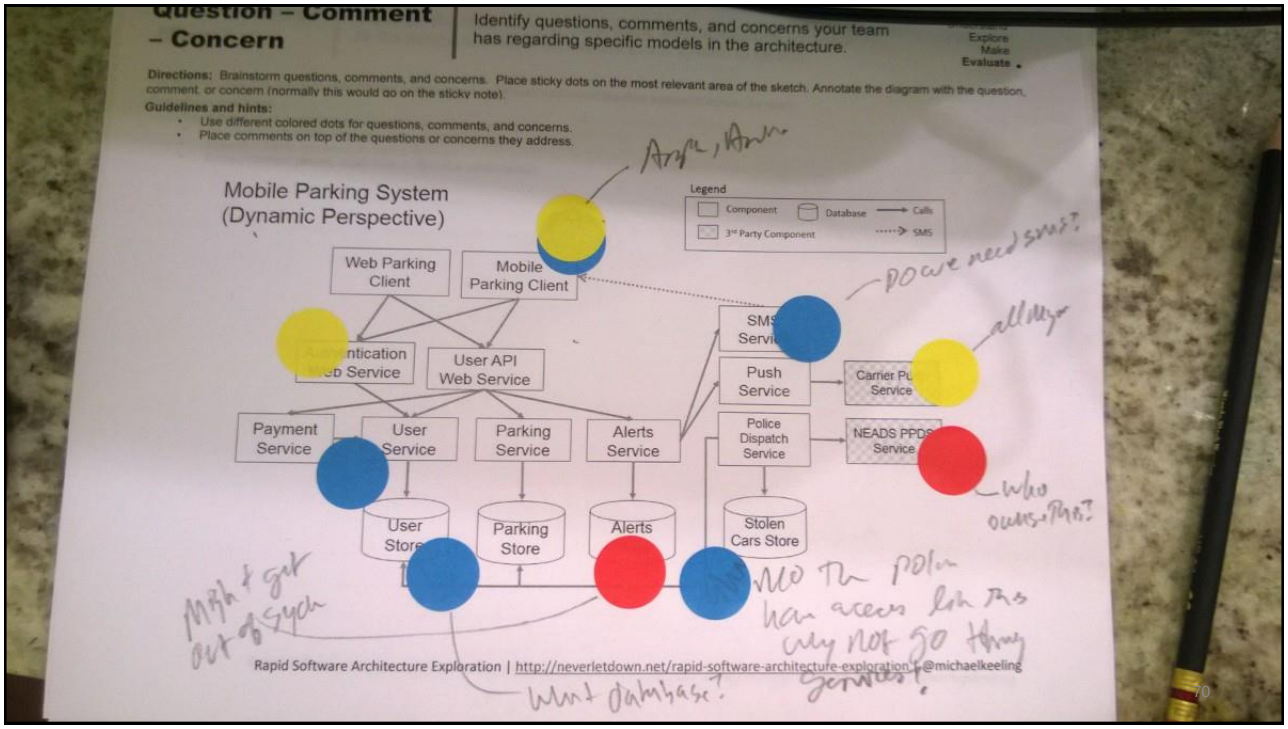
Participants Team, relevant stakeholders

67

Mobile Parking System (Dynamic Perspective)



69



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

Question | Comment | Concern

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.

71

Show and Tell
(Reflection)

72

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?

73

WRAP-UP

74

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,…

75

Three Fs of Good Design Methods

Fast

Effective

Fun

76

Silver Toolbox

77

Thank you!

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

Help create
IBM Watson

78