

# TIMG 5006

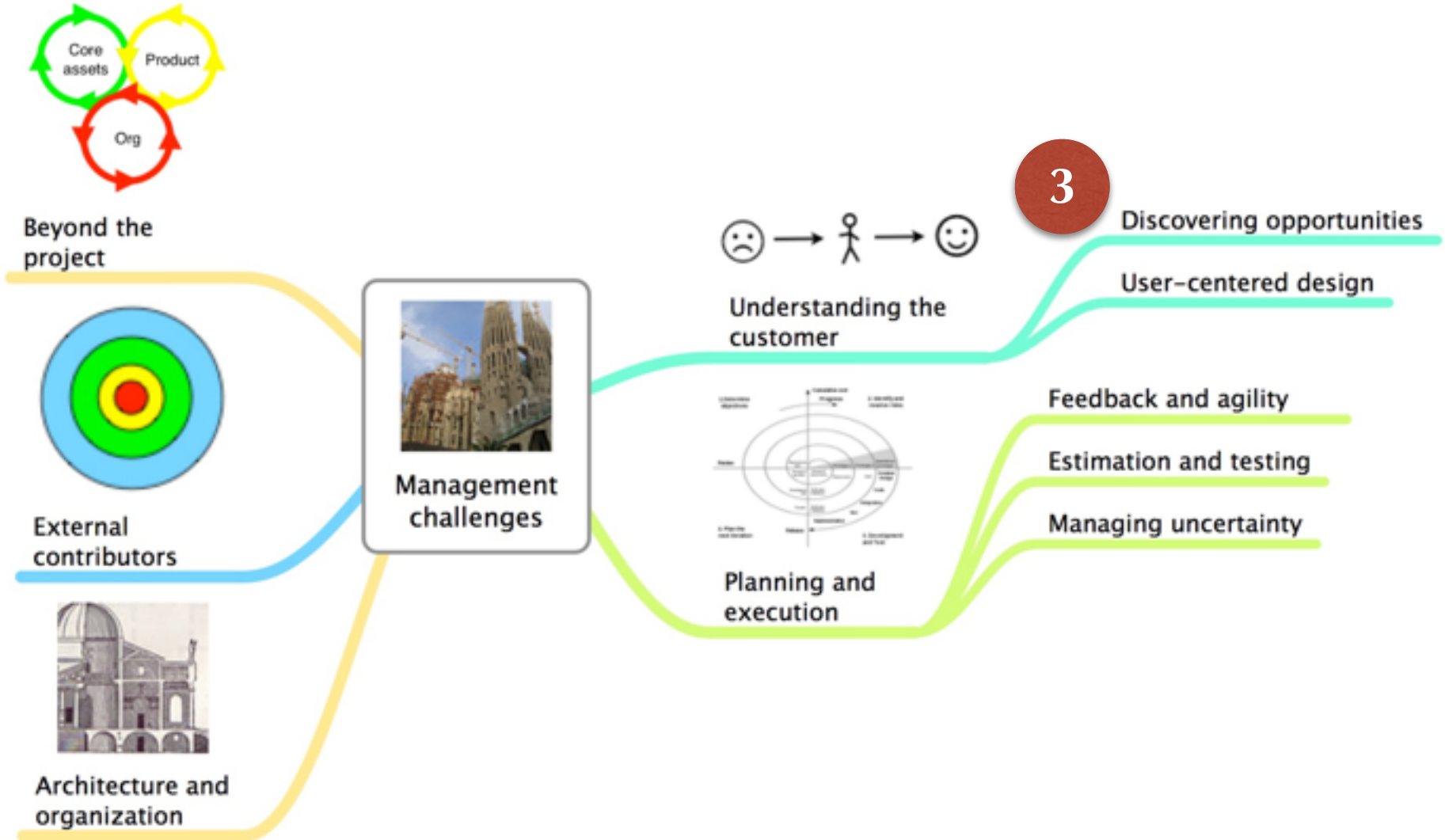
## Management of Software Engineering Projects

Session 3: Sep 16

**Michael Weiss**

[www.carleton.ca/tim](http://www.carleton.ca/tim)  
[www.carleton.ca/tim/tim.pdf](http://www.carleton.ca/tim/tim.pdf)

# Course topics



# Session 3 objectives

- Upon completion of this session, you will know about
  - Advantages of observation over questions
  - Tools and practices for discovering what the customer needs but doesn't know
- And you will be able to
  - Discover the customer's unstated needs
  - Describe the big picture for your software: what is the value it creates for your customers

# Agenda

1. Assignments
2. Adrenaline junkies
3. Understanding customers
4. Requirements and value
5. Key lessons
6. Questions

# 1. Assignments

- Group assignments
  - Write a project management pattern (20%)  
Post topic on wiki by **Sep 23**
  - Design a card / board game for training a software project management skill (40%)
- Class participation (10%)
- Exam (30%)

# Readings for Session 3

- Leonard, D. & Rayport, J. (1997), Spark innovation through empathic design, *Harvard Business Review*, 75, Nov-Dec, 102-115.
- Weiss (2012a), User frustrations as opportunities, *TIM Review*, April, <http://timreview.ca/article/546>
- Weiss (2012b), Creating Customer Value Propositions for Technology Products, *EuroPLoP*
- Lehtola, L., Kauppinen, M., Vähäniitty, J., & Komssi, M. (2009), Linking business and requirements engineering: is solution planning a missing activity in software product companies?, *Requirements Engineering*, 14(2), 113-128.

## 2. Adrenaline Junkies

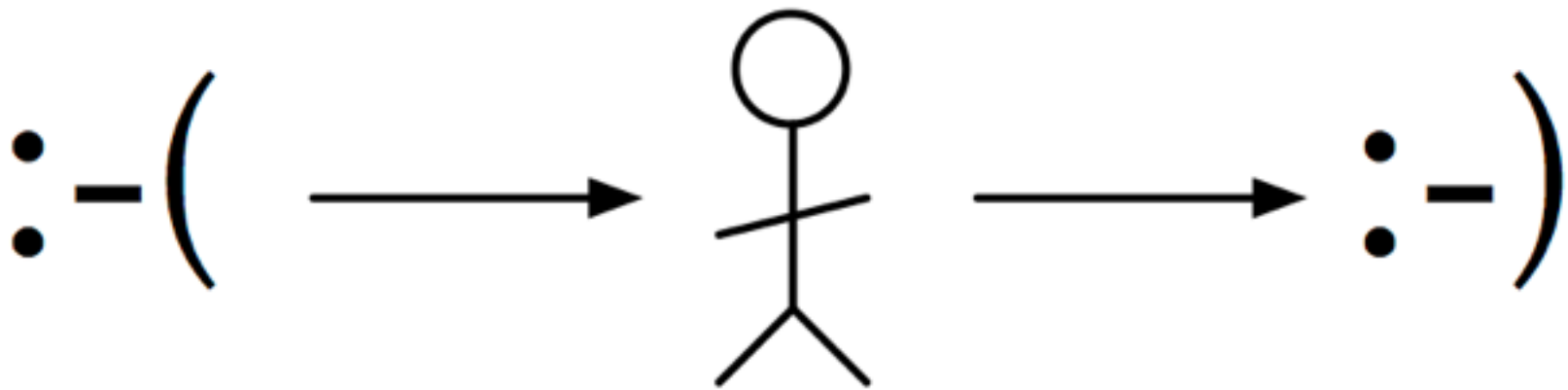
Feature Soup	Product has many features that don't address real customer needs
Straw Man	A straw man is an incomplete solution that allows you to get feedback from clients
Adrenaline junkies	Frenzied activity is mistaken as a sign of productivity. Telling signs: shifting priorities and manufactured urgency
Lewis & Clark	Upfront exploration of a business domain to discover potential opportunities

# Patterns (good, bad, and ugly)

Feature Soup	
Straw Man	
Adrenaline Junkies	
Lewis & Clark	



# 3. Understanding customers



# A. Unarticulated needs

- Ability of customers to guide development of **new products and services** is limited by their experience and ability to imagine what is technically possible
- When a product or service is **well understood**, interviews, focus groups, or surveys are great ways to get input from customers and influence them
- Existing products **frame** how how customers can articulate their needs. They may also be accustomed to work within the constraints of what is available



Myopia

# Don't ask, observe!

- For new products or services, a different approach is required based on observation: **empathic design**



# Asking vs observing

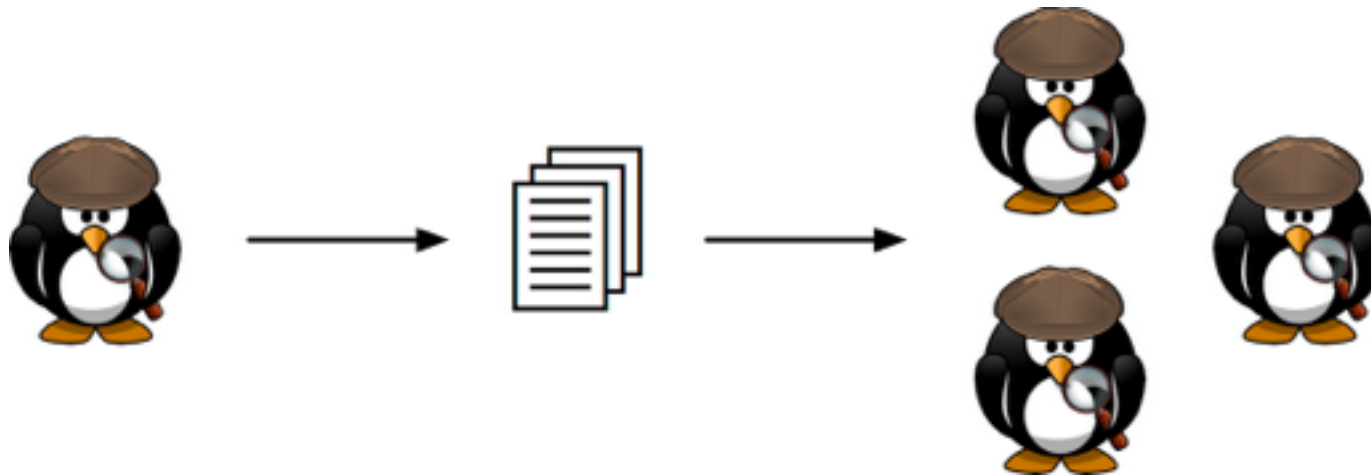
Asking	Observing
People can't ask for what they don't know can be built	
People are not good observers of their own behavior	
People like to please	
People are less likely to recall intangible aspects	
People's desires are framed by their experience	
Questions are often biased	
Questions interrupt usual flow	
Questions stifle opportunities	

# Empathic design

- Start with **observation** (who, where, how)
- Capture **data** (visual, auditory, and sensory clues)
- **Synthesize** your observation (needs, desires)

Who

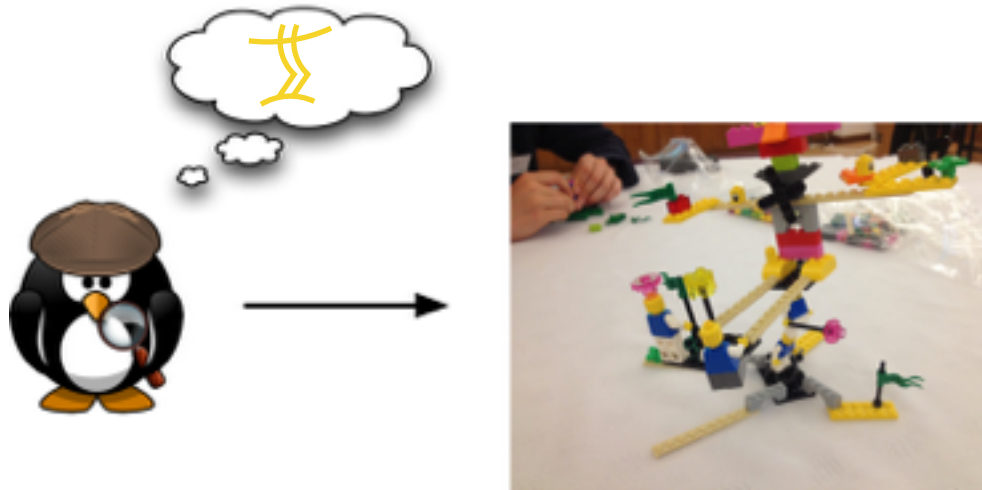
What



# Empathic design (more)

- Brainstorm for **solutions** (visual representations)
- Develop **prototypes** (possible solutions) that clarify concept for a new product or service
- Goal is to discover what users “**have not asked for**”

How



# Learning by observation

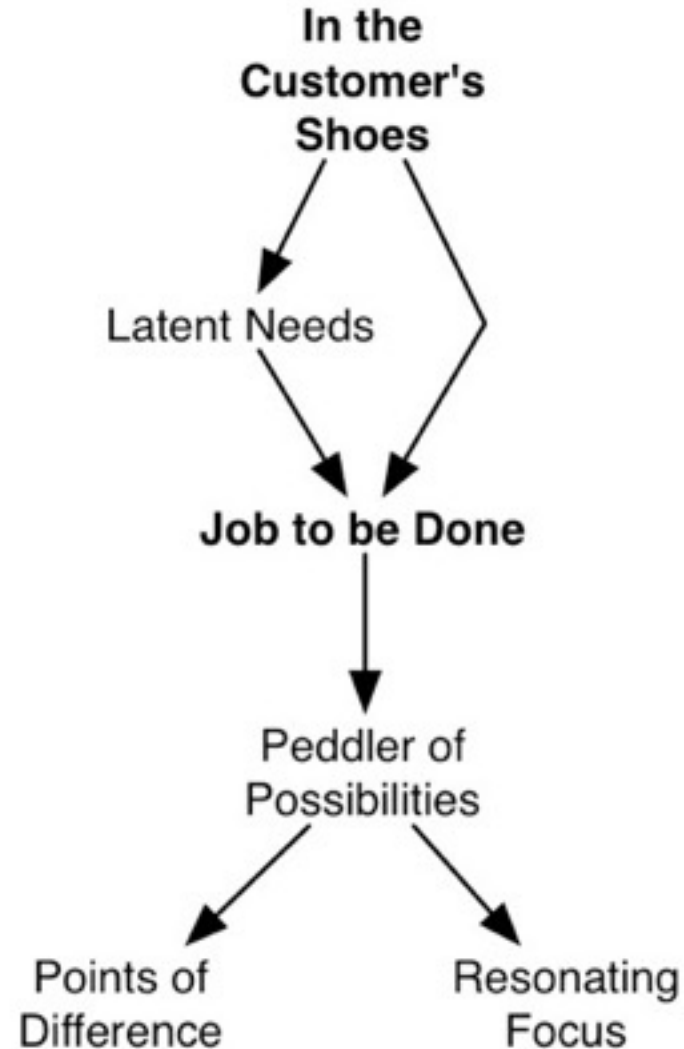
- What prompts users to use the product? Products or services may be used in **unexpected ways**
- How does **product fit** into the users' environment, eg Intuit's Follow Me Home program
- Do users **redesign** your product to serve their own needs? These are "edge" cases/early adopters
- What **intangible attributes** do customers like?
- What **breakdowns** do customers encounter when using your products that they don't know can be fixed?

## B. Learning from user frustrations

- What **problem** are you solving for your user/customer?
- What **frustrations** do users experience with current solutions? Can they articulate their frustrations?
- How are **users solving** their problem today? What do they actually want to achieve?
- What **better ways** are there to solve the user's problem? Do you have the required skills?
- How is **your solution** different from other solutions on the market (points of difference)?



- What customer problem are you solving?
- Are customers able to articulate their needs?
- What is it that customers actually want to achieve?
- What are better ways of solving the customer's problem?
- What are the points of difference of your solution?



# In the customer's shoes

- What **customer problem** you are you solving?
  - Your perception of what problems customers face and their priority may not be accurate
  - However, just asking customers will only tell you about the problems customers can articulate
  - Asking customers about their needs will only lead to incremental improvements (myopia!)
- Learn what problems customers face, but **look beyond** needs that customers can readily identify
  - Take the customer's perspective: your product must address needs the customer has

# Latent needs

- Customer have perceived needs and latent needs
- Perceived needs are needs user can articulate
- **Latent needs** are expressed through frustrations for which the user does not have a solution
- To observe latent needs, you need to observe users in their target setting (“get out of the building”)
- This is often done by “shadowing” users

- What **task** are customers trying to achieve?
  - Customers are already solving their problem somehow, but not always efficiently
  - Customers have access to solutions beyond those offered by your direct competition
- Identify the “**job to be done**”
  - What job would customers “hire” your solution for
  - What alternative solutions are available to them: those are the solutions you compete against

# Example

- A customer's job-to-be-done might be to

have access to a library  
of content



not own a portable device  
for reading books



# Exercise

- Assume you are designing a portable scanner
- Please form groups, discuss for 10 min how you will do this, then report back to the class

# Exercise

- Assume you are designing a portable scanner
- Please form groups, discuss for 10 min how you will do this, then report back to the class



# Don't act on perceived needs!

- Assume your customer is an industrial designer and you observed them “scanning” objects they find
- What is their job-to-be-done? What frustrates them?





# Observe, then build!

Problem	
User frustrations	
Solutions available to users & JTBD	
Your solution	
Points of difference	

## 4. Requirements and value

- What is a requirement? A property that a product must have to **provide value** to a stakeholder!
- **Market-driven** vs **custom** software development
- Companies need to implement the **most valuable requirements** in each release of the software
- However, in order to **prioritize** requirements, we need to understand the business value created
- Planning must also encompass **market-oriented** roadmaps, not only short term product ones

# Solution planning

Customer segments	Identify <b>types of customers</b> and describe their main <b>characteristics</b> and/or <b>jobs to be done</b>
Value creating activities	Identify <b>activities</b> customers perform relating to the solution (context for the solution)
Solution	Identify <b>components</b> of the solution concept (software, services, training, etc.)
Value propositions	Describe <b>main reason</b> why customers/end-users want to buy/use this solution
Message	Key <b>message</b> that summarizes the business rationale of the solution for stakeholders

# 5. Key lessons

---



## 6. Questions

- What problem does user-centered design solve?
- If we think of software product development as an investment, what are the implications?

# Readings for next session

- Beyer, H., Holtzblatt, K., & Baker, L. (2004), An agile user-centered method: Rapid contextual design, *Extreme Programming and Agile Methods – XP/Agile Universe*, LNCS 3134, 527-554, Springer.
- Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Perrson, J., & Cajander, Å. (2003), Key principles for user-centred systems design, *Behaviour and Information Technology*, 22(6), 397-409.
- Martin, A., Biddle, R., & Noble, J. (2009), XP customer practices: A grounded theory, *Agile Conference*, 33-40, IEEE.
- TBD

# Credits

- All clipart from [openclipart.org](http://openclipart.org), unless noted otherwise
- Lego, author's own photo