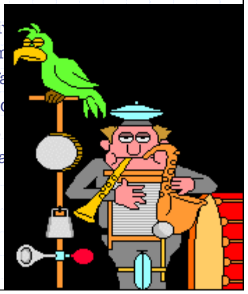
Categories of MM Applications

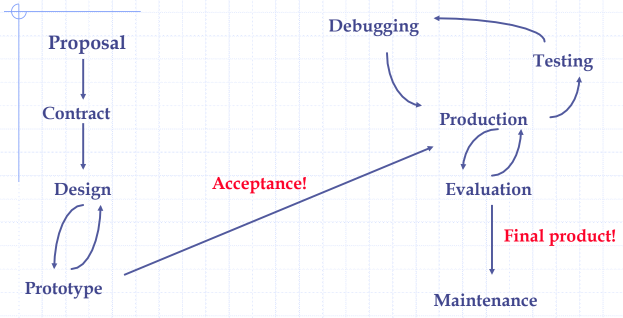
- Advertising
  - Presentations
  - Marketing tools
- Education
  - Courseware
  - Kiosks
  - Simulations
- Edutainment
  - Consumer titles
  - Kids’ books
- Varia
  - Custom productions

Plurality of MM Development Roles

- Management
  - Executive Producer
  - Producer
  - Project Manager
- Design
  - Creati
  - Inform
  - Interfa
- Productic
  - Artist
  - Progra



Development Process



Proposal

Contract

Design

Prototype

Acceptance!

Debugging

Testing

Production

Evaluation

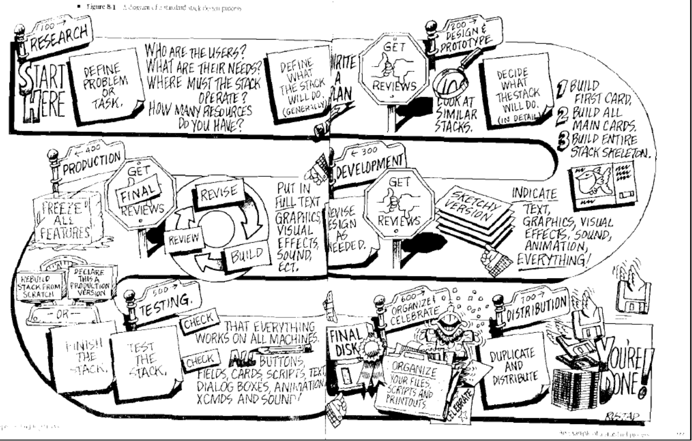
Final product!

Maintenance

But it is not the whole story…

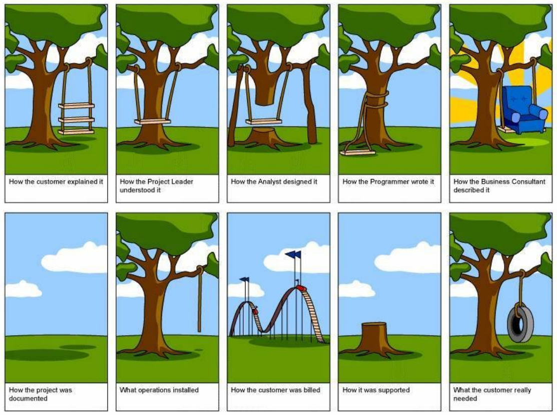
Let’s play Design!

Figure 8-1 A diagram of a standard stack design process



100 →
RESEARCH
START HERE
DEFINE PROBLEM OR TASK.
WHO ARE THE USERS?
WHAT ARE THEIR NEEDS?
WHERE MUST THE STACK OPERATE?
HOW MANY RESOURCES DO YOU HAVE?
DEFINE WHAT THE STACK WILL DO. (GENERALLY)
WRITE A PLAN
GET REVIEWS
200 →
DESIGN & PROTOTYPE
LOOK AT SIMILAR STACKS.
DECIDE WHAT THE STACK WILL DO. (IN DETAIL)
1 BUILD FIRST CARD.
2 BUILD ALL MAIN CARDS.
3 BUILD ENTIRE STACK SKELETON.
← 300
DEVELOPMENT
INDICATE TEXT, GRAPHICS, VISUAL EFFECTS, SOUND, ANIMATION, EVERYTHING!
SKETCHY VERSION
GET REVIEWS
REVISE DESIGN AS NEEDED.
PUT IN FULL TEXT GRAPHICS, VISUAL EFFECTS, SOUND, ECT.
REVISE
BUILD
REVIEW
GET FINAL REVIEWS
← 400
PRODUCTION
FREEZE ALL FEATURES
REBUILD STACK FROM SCRATCH
DECLARE THIS A PRODUCTION VERSION
- OR -
500 →
TESTING.
FINISH THE STACK.
TEST THE STACK.
CHECK THAT EVERYTHING WORKS ON ALL MACHINES.
CHECK ALL BUTTONS, FIELDS, CARDS, SCRIPTS, TEXT, DIALOG BOXES, ANIMATION, XCMDS AND SOUND!
FINAL DISK
600 →
ORGANIZE & CELEBRATE
ORGANIZE YOUR FILES, SCRIPTS AND PRINTOUTS.
CELEBRATE
700 →
DISTRIBUTION
DUPLICATE AND DISTRIBUTE
YOU'RE DONE!
RUSTAD

0. Who needs design?



How the customer explained it

How the Project Leader understood it

How the Analyst designed it

How the Programmer wrote it

How the Business Consultant described it

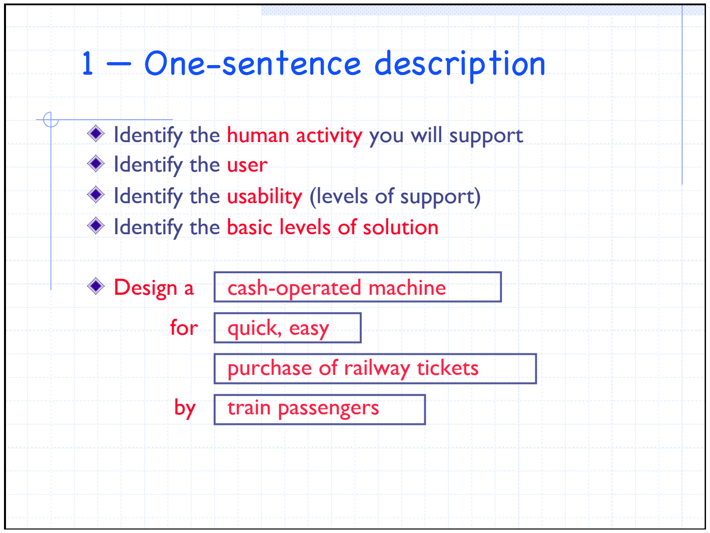
How the project was documented

What operations installed

How the customer was billed

How it was supported

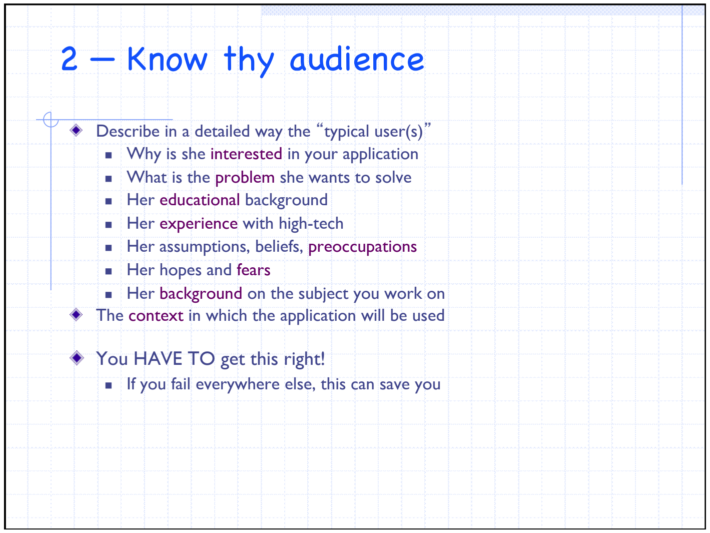
What the customer really needed



1 — One-sentence description

- Identify the human activity you will support
- Identify the user
- Identify the usability (levels of support)
- Identify the basic levels of solution

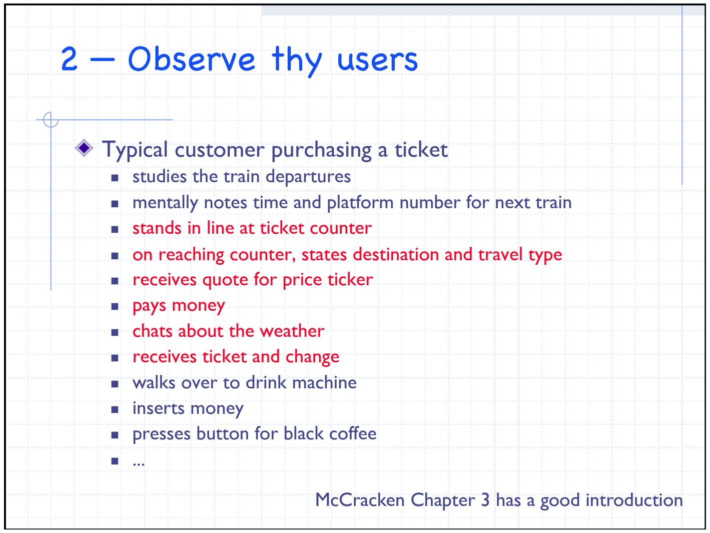
- Design a cash-operated machine
  for quick, easy
  purchase of railway tickets
  by train passengers



2 — Know thy audience

- Describe in a detailed way the “typical user(s)”
  - Why is she interested in your application
  - What is the problem she wants to solve
  - Her educational background
  - Her experience with high-tech
  - Her assumptions, beliefs, preoccupations
  - Her hopes and fears
  - Her background on the subject you work on
- The context in which the application will be used

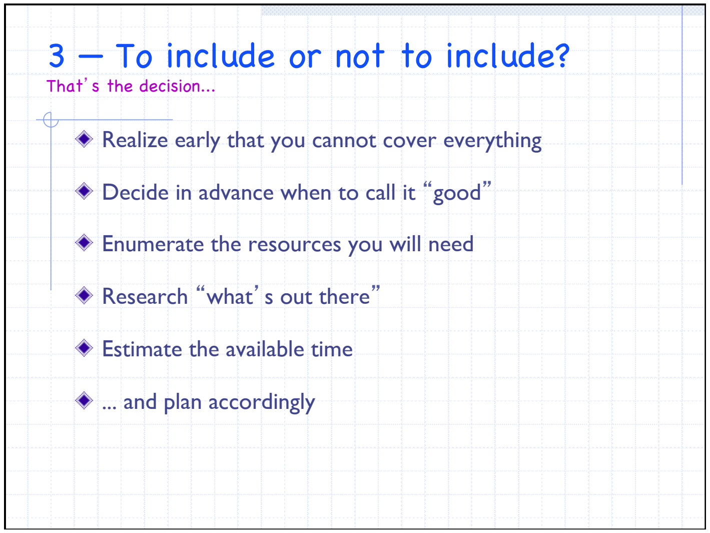
- You HAVE TO get this right!
  - If you fail everywhere else, this can save you



2 — Observe thy users

- Typical customer purchasing a ticket
  - studies the train departures
  - mentally notes time and platform number for next train
  - stands in line at ticket counter
  - on reaching counter, states destination and travel type
  - receives quote for price ticker
  - pays money
  - chats about the weather
  - receives ticket and change
  - walks over to drink machine
  - inserts money
  - presses button for black coffee
  - …

McCracken Chapter 3 has a good introduction



3 — To include or not to include?

That’s the decision…

- Realize early that you cannot cover everything
- Decide in advance when to call it “good”
- Enumerate the resources you will need
- Research “what’s out there”
- Estimate the available time
- … and plan accordingly

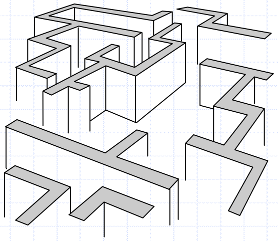
4 — Find a good metaphor

- … "real world" object, event, thing
  that your audience is familiar with
- Operating System: office desktop (MacOS)
- Electronic mail: air mail (Eudora)
- Painting: painter's palette (Fractal Designer)
- Database: organizing cards (HyperCard)
- Writing: typewriter (Word)
- Animation: theatrical stage (Director)

More on metaphores later…

5 — Make it easy to navigate

- If it is difficult to use,
  it will not be used (simple, huh?)
- User needs to know:
  - where am I?
  - where have I been?
  - who am I?
  - where can I go from here?
  - how can I get there?
  - ~~is there life after death? ;-)~~
- Care about Usability



More on Navigation later…

6 — Encourage interaction. Early!

- Get them to do something early on, not just sit back and watch
  — click, type, select …
- Think of the novice user
  - Give them a short, well laid-out, to the point, introduction
- Think of the advanced user
  - give them a way to skip introduction
- Specialized help for every screen
- Follow Chinese wisdom:
  - I hear and forget;
    I see and remember;
    I Do and Understand!
- KISS!!

7 — Don't get it original; get it right!

- Design screen grids
- Be consistent in the screen design
- Minimize administrative debris
- Use icons
- Brief text — reading from screen is difficult
- Scrolling is evil
- Try serifs and non-serifs
- Use meaningful lines
- Be forgiving
- Break the rules,
  but only when you know how and why you follow them

"Talent imitates, but genius steals" T.S.Eliot

8 — THINK before implementing

(but DO change to improve)

- Novices always flunk this…
- Don't be defensive; don't be romantic
- The good news: The second time is easier
  - … and the third, a piece of cake!
- Keep documentation!! (or waste your time)
- Keep versions! (or mess it up and lose it)
- When you think you are done, either
  - redo the whole think from scratch (recommended)
  - declare the current version "beta"

9 — Test, evaluate, test, test, test

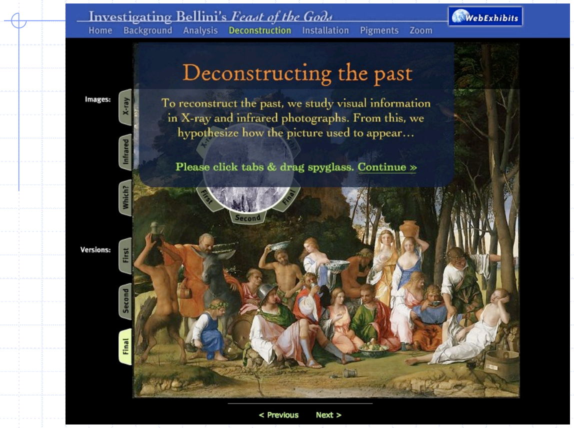
- Test at several times of production
- Ultimate testers are typical users
- Listen; don't argue
- Watch testers quietly;
  their navigating experience says a lot
  - Do they get confused easily?
  - Do they use help often?
  - Do they find the solution quickly?

McCracken Chapter 8 explores testing

10 — The Finishing Touches

- Checked for typos?
  grammatical errors?
- Do text & graphics match?
- All buttons work?
- All scripts work
  without crashing?
- Do alert & dialog boxes work?
- Provides forgiving/cancel options?
- Do links connect correctly?
- Screen transition consistent?
- Is video/sound/animation timing acceptable?
- Does it work across platform?
- Needs installation on various machines?
- Obtain copyrights? legal notices? logos?
- What is the smallest & best machine configuration?
- Need accompanying extensions? other s/w?
- Need folder placement? aliases placed?
- Checked for VIRUSES?
- registration? installation note?
- "read me first" file?
- Has supporting WWW site?

Case Study



Deconstructing the past

To reconstruct the past, we study visual information in X-ray and infrared photographs. From this, we hypothesize how the picture used to appear…

Please click tabs & drag spyglass. Continue »