HUMAN-COMPUTER INTERACTION

Lecture 21-22 – Goal-Directed Design

BIT 10 CD
Fall 2011

Hamid Mukhtar, SEECS-NUST

Introduction

- If we design and construct products in such a way that the people who use them achieve their goals, these people will be satisfied, effective, and happy and will gladly pay for the products and recommend that others do the same
- Make people happy, and your products will be a success

What's missing?

- Why then are so many digital products so difficult and unpleasant to use?
- Why aren't we all happy and successful?
- Developers end up creating technologically focused solutions that are difficult to use and control
- Instead of planning and executing with a mind towards satisfying the needs of the people who purchase and use their products

Developer's perspective

- Good developers are focused on solving challenging technical problems, following good engineering practices, and meeting deadlines
- In this pursuit, they forget the human factor and the real value of the product
- They rarely take into account the users' goals, needs, or motivations, and at the same time tend to be highly reactive to market trends and technical constraints
- Result in products that lack a coherent user experience

Evolution of SDP

- The early days of the software industry when smart programmers dreamed up products, and then built and tested them

- Professional managers were brought in to help facilitate the process by translating market opportunities into product requirements

- The industry matured, testing became a discipline in its own right, and with the popularization of the graphical user interface (GUI), graphic designers were brought in to create icons and other visual elements

- The Goal-Directed approach to software development where decisions about a product's capabilities, form, and behavior are made before the expensive and challenging construction phase

Don’t be rude

- They often blame users for making mistakes that are not their fault

- Why didn’t the program notify the library? What did it want to notify the library about? Why is it telling us? And what are we OKing, anyway? It is not OK that the program failed!

Don’t require people to think

- Digital products regularly assume that people are technology literate
- How can I rename this presentation while it’s open?
- Programs often express themselves in incomprehensible jargon that cannot be understood by normal users
  - “Please specify the Mac Address of your PC”

Don’t exhibit poor behavior

- For example, after Windows get automatic updates, it may restart without the user’s intention
- Older versions of MS Word would interrogate you to save the file after printing even if you had not made any modifications

Don’t require humans to do heavy lifting

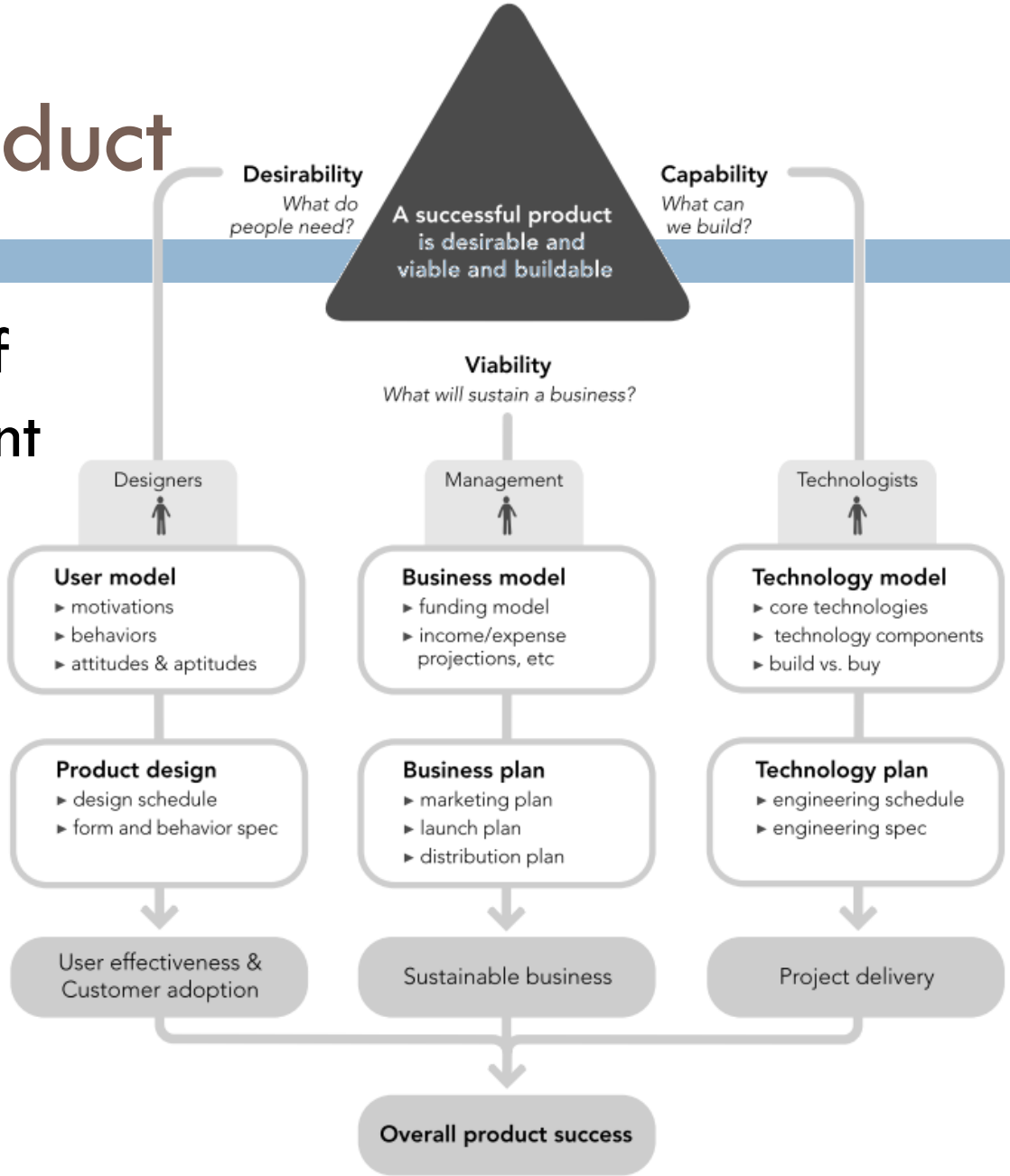
- How easy is it to move text from your notes to SMS in Nokia or HTC or Samsung phones?

Why are these products so bad?

- Ignorance about users
  - why they are doing whatever it is they might need our product for, why they might want to choose our product over our competitors, or how we can make sure they do?
- A conflict of interest between serving human needs and construction priorities
  - programmers' performance is typically judged by their ability to code efficiently and meet incredibly tight deadlines. They prefer ease of coding to ease of use
- The lack of a process for understanding human needs as an aid to developing appropriate product form and behavior.

Successful product

- The modern triad of product development concerns identified by Larry Keeley

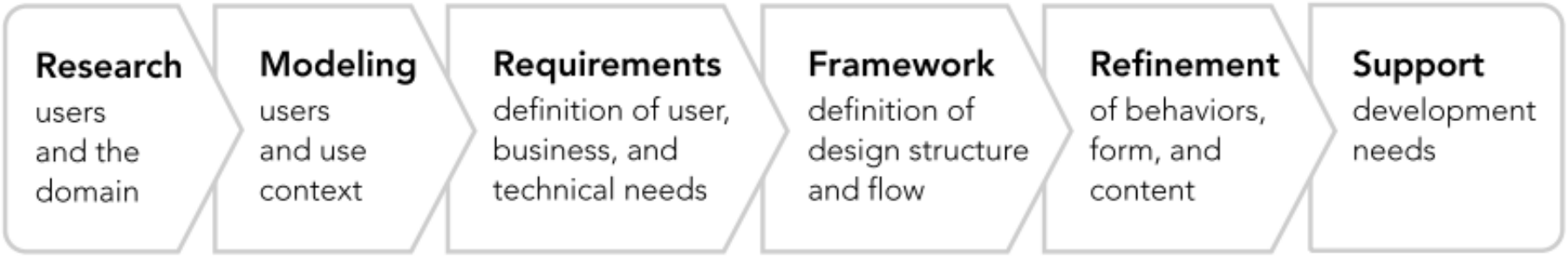


Goal-Directed Design

Recognizing User Goals

- So what are user goals?
- How can we identify them?
- How do we know that they are real goals?
  - rather than tasks they are forced to do by poorly designed tools or business processes?
- Are they the same for all users?
- Do they change over time?

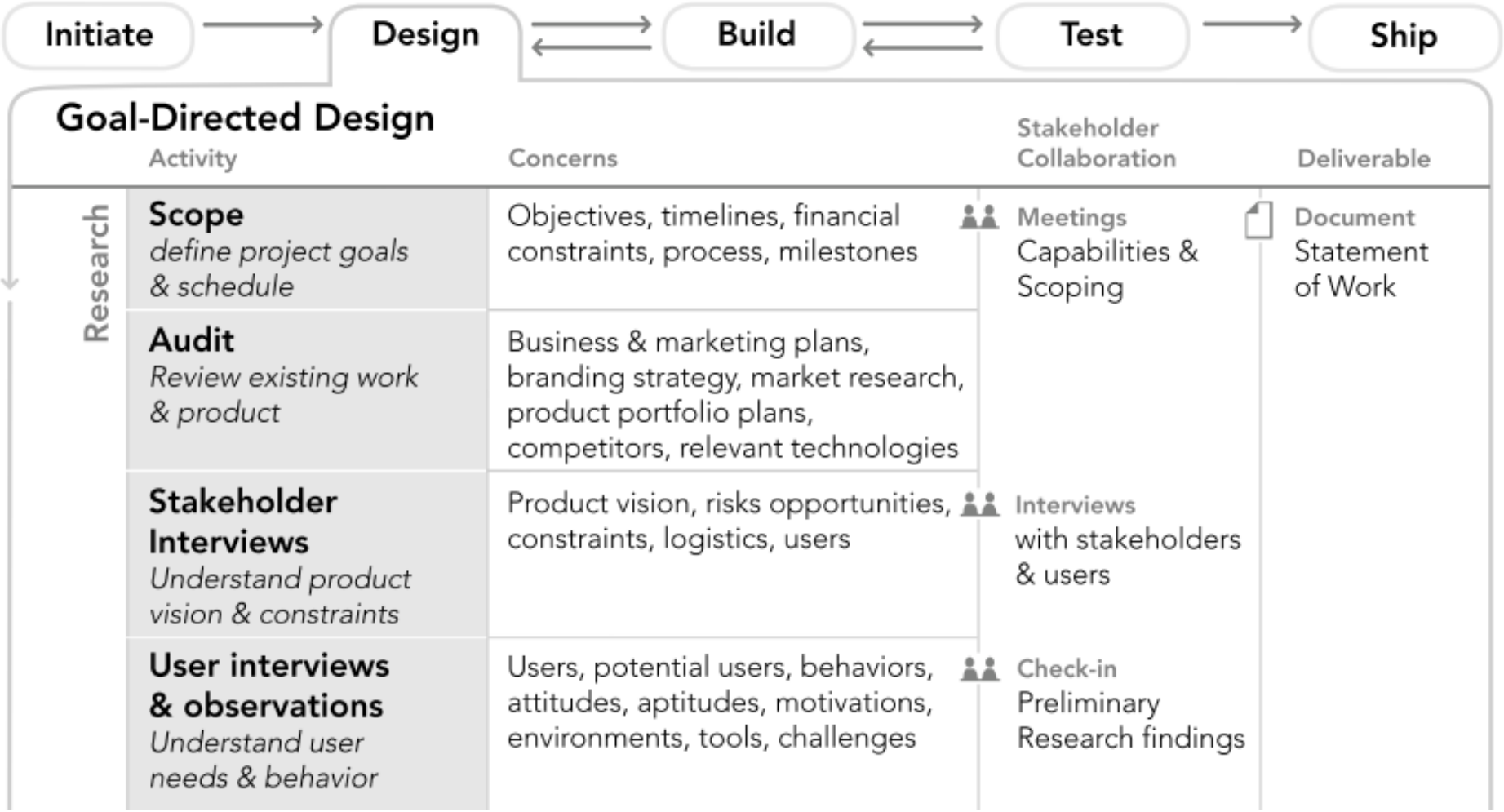
GDD Process


Research
users and the domain
Modeling
users and use context
Requirements
definition of user, business, and technical needs
Framework
definition of design structure and flow
Refinement
of behaviors, form, and content
Support
development needs

Research

- Employs ethnographic studies (observation and contextual interviews) to provide qualitative data about potential and/or actual users of the product
- Competitive product audits, reviews of market research and technology white papers and brand strategy
- One-on-one interviews with stakeholders, developers, subject matter experts (SMEs), and technology experts as suits the particular domain

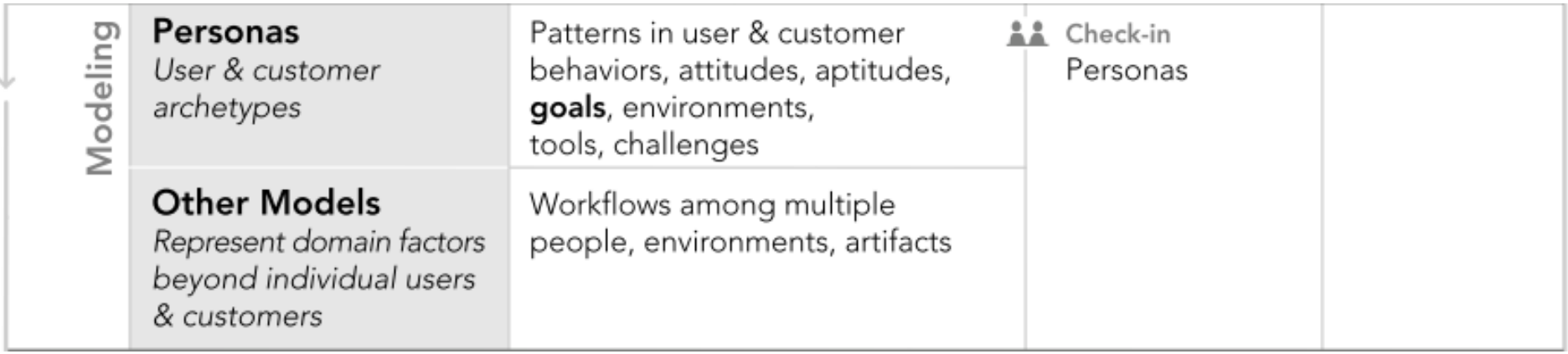
Research


Initiate
Design
Build
Test
Ship
Goal-Directed Design
Activity
Concerns
Stakeholder Collaboration
Deliverable
Research
Scope
define project goals & schedule
Objectives, timelines, financial constraints, process, milestones
Meetings
Capabilities & Scoping
Document
Statement of Work
Audit
Review existing work & product
Business & marketing plans, branding strategy, market research, product portfolio plans, competitors, relevant technologies
Stakeholder Interviews
Understand product vision & constraints
Product vision, risks opportunities, constraints, logistics, users
Interviews
with stakeholders & users
User interviews & observations
Understand user needs & behavior
Users, potential users, behaviors, attitudes, aptitudes, motivations, environments, tools, challenges
Check-in
Preliminary Research findings

Modeling

- During the Modeling phase, behavior and workflow patterns are synthesized into domain and user models
- Domain models can include information flow and workflow diagrams
- User models, or personas, are detailed, composite user archetypes that represent distinct groupings of behaviors, attitudes, aptitudes, goals, and motivations observed and identified during the Research phase
- Personas are then used in the scenario-based approach to design

Modeling


Modeling
Personas
User & customer archetypes
Patterns in user & customer behaviors, attitudes, aptitudes, goals, environments, tools, challenges
Check-in
Personas
Other Models
Represent domain factors beyond individual users & customers
Workflows among multiple people, environments, artifacts

Requirements Definition

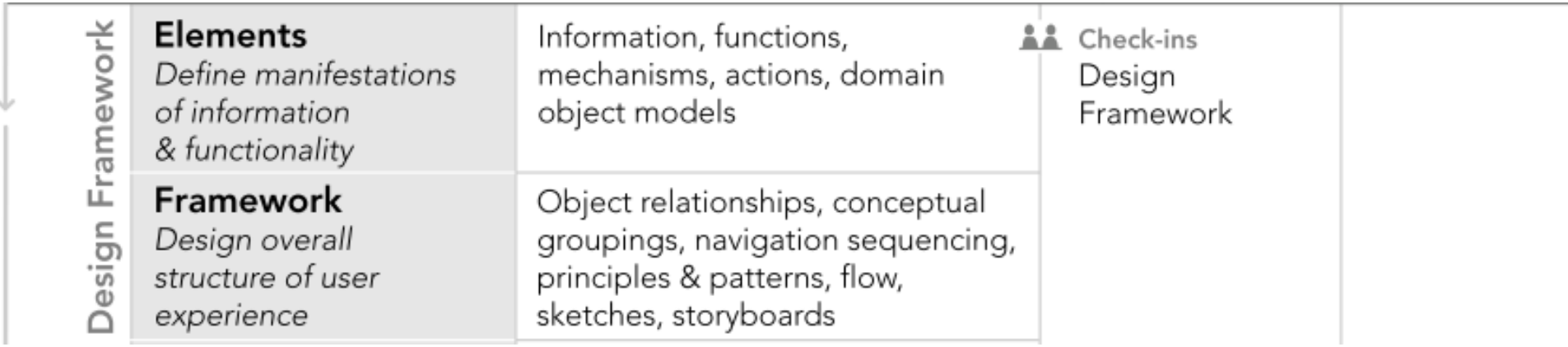
- This phase employs scenario-based design methods focusing the scenarios not on user tasks
- Personas provide an understanding of which tasks are truly important and why, leading to an interface that minimizes necessary tasks (effort)
- Personas become the main characters of these scenarios
  - Designers explore the design space via a form of role-playing
  - Designers consider the personas' skills and physical capabilities as well as issues related to the usage environment

Requirements Definition

Framework Definition

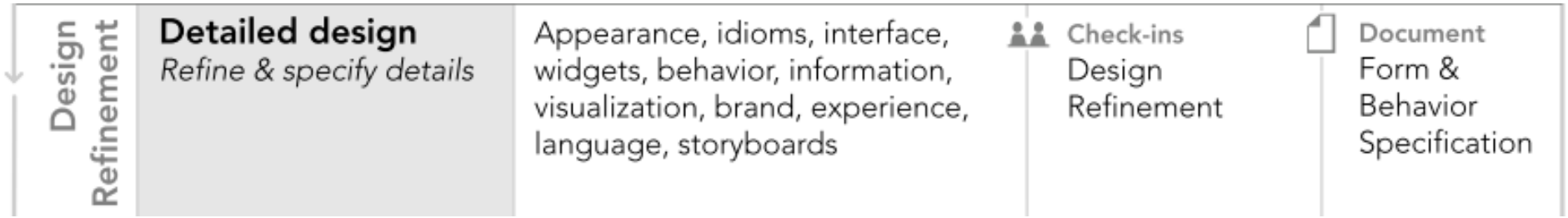
- In this phase, designers create the overall product concept, defining the basic frameworks for the product's behavior and visual design
  - *Frameworks outline possible courses of action or present a preferred approach to an idea or thought*
- Interaction framework
  - Determine appropriate system behavior in a variety of contexts
- Visual framework
  - Overall interface structure to develop options for typography, color palettes, and visual style

Framework Definition



Refinement

- Similar to the Framework Definition phase, but with increasing focus on detail and implementation
- Form and behavior specification



Development Support

- Any design solution can't possibly anticipate every development challenge and technical question
- It's important to be available to answer developers' questions as they arise during the construction process
- The design may be adjusted as the development team prioritizes their work and makes trade-offs to meet deadlines
- If the interaction design team is not available to create these solutions, developers are forced to do this under time pressure, which has the potential to gravely compromise the integrity of the product's design

Development Support

Conclusion

- Goals, not features, are the key to product success
- The successful interaction designer focuses on users' goals amid the pressures and chaos of the product-development cycle
- GDD makes collaboration with developers and businesspeople easier, and ensures that the design in question isn't guesswork, the whim of a creative mind, or just a reflection of the team members' personal preferences

Questions you should ask?

- Who are my users?
- What are my users trying to accomplish?
- How do my users think about what they're trying to accomplish?
- What kind of experiences do my users find appealing and rewarding?
- How should my product behave?
- What form should my product take?
- How will users interact with my product?
- How can my product's functions be most effectively organized?

Questions you should ask?

- How will my product introduce itself to first-time users?
- How can my product put an understandable, appealing, and controllable face on technology?
- How can my product deal with problems that users encounter?
- How will my product help infrequent and inexperienced users understand how to accomplish their goals?
- How can my product provide sufficient depth and power for expert users?