Cheating Policy

• READ THE WEB SITE
  • http://cs.nyu.edu/courses/spring05/G22.2280-001/cheating_policy.htm

• Cheating Gets You an "F"
  – I take cheating very seriously.
  – **If you cheat you will get an "F" for the course.**
  – Unless I tell you that doing an assignment together is required or optional, **assume that you must DO YOUR OWN WORK.**
  – The goal of enforcing a **no cheating** policy is to insure that you learn the material presented in the course and earn the grade you get. All forms of cheating defeat the goal of learning.
  – During the course I use several **anticheating technologies** including running content analyzers on submitted homework and several versions of assignments/tests.
Specific Examples Of Cheating

• The following constitute forms of cheating (some but not all forms):
  – **Getting or giving** your assignments to another person. Using assignments submitted in previous semesters
  – **Soliciting** to have someone do your assignments in part/or in whole for you.
  – **Having someone** do your assignments in part/or in whole for you
  – **Doing someone** else's assignments for them.
  – **Copying** someone else's work that includes getting stuff from the web and other sources, not just your classmates
  – Submitting work that is not **completely** created by you
  – **Looking** at someone else's test during an exam or asking someone during the exam for help. Possessing an unauthorized copy of an exam.
  – **Using technology** to get answers during an exam. Telling someone, in detail, what answers to enter on their assignments.
  – **Knowing that someone has cheated** and NOT telling me about it. I know this maybe difficult for some to comply with but I expect that if you know you will encourage those involved to either redo the work legitimately or to inform me that they cheated. Withholding this information from me is aiding and abetting cheating and thus you too are cheating.
  – The persons **giving** the information and those **using it** are **BOTH** cheating and will **BOTH** receive an "F" in the course.
• **Helping Each Other**
  – I encourage you to help each other to learn but NOT to give the answers. One way to help is to discuss the problem (if permitted in the assignment) but not give the other person YOUR solution. Mentoring means guiding them to find the solution themselves NOT giving them the answers.

• **A Quick Cheating Detector**
  – A quick test if you cheated or not is if you understand how the solution was arrived at and could create a new solution, on your own, for a related problem. If you can't answer that with a firm "yes" and you received information/help from others, it's possible you have cheated. If you had done the work yourself you'd understand how you arrived at the solution.

• **Please don't cheat, it makes my job much harder and it mainly cheats you of the education that you are paying so much for. If you feel the need to cheat, come talk to me or the TA about why and we will find an honorable solution.**
Design Process

An iterative process for designing great UIs
Process

• Steps and activities
• Executed in a sequence
• Iterative means we may revisit, extend, rework, restart and redo an activity later
• Loosely based on the Unified Process (UP) –
  http://www.agilemodeling.com/essays/agileModelingRUP.htm
UP - Unified Process
(a collection of public domain ideas about how to structure a project so it doesn’t fail)
Parallel Workflows (Disciplines)

- Information Gathering
  - “Gathering information about the problem and constraints of the solution”
- Analysis
  - “Problem Definition”
- Design
  - “Solution Creation”
- Implementation
  - “Implementing The Proposed Solution”
- Testing & Fixing
  - “Verification and Validation of the Solution”
- Training
  - “Learning to Use the Solution”
- Documentation
  - “Creating Manuals and Descriptions of the Solution Implemented”
- Packaging
  - “Packaging Solution for Distribution and Installation”
- Managing the Project
  - “Coordination, Oversight, Facilitation, Tracking and Resource Management”
- Deployment
  - “Putting the solution in the Hands of The Users in the Field”
Different Phases

• Inception (I)
  – Goals
  – define the general business case for the project
  – Define the stakeholders
  – Get a vague idea of the problem being solved (opportunity) and the solution possibilities

• Elaboration (E)
  – Building a prototype of the application – more structural than pretty

• Construction (C)
  – Building the parts of the application, testing them, fixing the bugs.

• Transition (T)
  – Getting the application ready to run in production.
  – Packaging, documentation, training, …

• Production (P)
  – Real users using it and in the process needing support, finding hidden bugs and requesting new features
Relative amount of work in each phase v. discipline

Disciplines

- Information Gathering
- Analysis
- Design
- Implementation
- Testing & Fixing
- Training
- Documentation
- Packaging
- Managing the project

Phases

- Inception
- Elaboration
- Construction
- Transition
- Production

As Time Progresses
Iterations

• The complete set of disciplines done in each iteration but with different levels of emphasis.
• Each PHASE is one or more iterations.
  – I 1 iteration
  – E 1 to 3 iterations
  – C 1 to 10 or more iterations
  – T 1 to as many minor releases as needed
  – P 1 until decision to do a new major release
• Iterations are usually between 1 week and 6 weeks. Average around 4 weeks (except inception.)
## Possible Iterations Per Phase

<table>
<thead>
<tr>
<th>Disciplines</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Gathering</td>
<td>Inception</td>
</tr>
<tr>
<td>Analysis</td>
<td>Elaboration</td>
</tr>
<tr>
<td>Design</td>
<td>Construction</td>
</tr>
<tr>
<td>Implementation</td>
<td>Transition</td>
</tr>
<tr>
<td>Testing &amp; Fixing</td>
<td>Production</td>
</tr>
<tr>
<td>Training</td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
</tr>
<tr>
<td>Packaging</td>
<td></td>
</tr>
<tr>
<td>Managing the project</td>
<td></td>
</tr>
</tbody>
</table>

As Time Progresses
We are initially going to focus on

• Inception phase
• Disciplines
  – Information Gathering
  – Analysis
• Later in the course we will change our focus to
  – Elaboration Phase
  – Analysis
  – Design
  – Testing
Focus for the Course

Disciplines

- Information Gathering
- Analysis
- Design
- Implementation
- Testing & Fixing
- Training
- Documentation
- Packaging
- Managing the project

Phases

<table>
<thead>
<tr>
<th>Phases</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
<th>Production</th>
</tr>
</thead>
</table>

As Time Progresses
Information Gathering

“Gathering information about the problem and constraints of the solution”

• Gathering information and specific requirements

• Techniques
  – Focus Groups of Users
  – Individual Interviews with Users
  – Observations of Users Doing Their Jobs
  – Workflow, Task and Business Process Review
  – Reviewing Existing Systems
  – Document review and business knowledge research
  – Use Cases & Scenarios
    • Describe, in Text How a User Would Use a System to Accomplish a Goal
Analysis
“Problem Definition”

• Abstraction & Generalization
  – Workflow, Actors/Roles, Tasks, Goals, Activities, Business Documents/Objects
• Documenting Functional and Non-functional Requirements & Constraints
• Generation of questions and clarifications of the problem domain
• Techniques
  – **Glossary of terms creation**
  – **User Analysis**
    • User Goals definition
    • User type identification / classification (taxonomy)
    • Task Analysis
    • Use case creation
  – **Business Domain Analysis**
    • Business entities (objects)
    • Identifying business processes
    • Information Analysis
  – **Requirements Definition**
Design
“Solution Creation”

• Solution Creation
  – What Does the software “look like” to provide the functionality that the users require?
• Design Patterns to Use
  – UI – Workbook, Wizard, Menued, Portal, Command, …
  – Non-UI – Command, Singleton, Tiered, MVC, SOA, …
• Platform Choice
  – Desktop GUI, Commandline, Web GUI, Voice, …
• Dev Language(s) & Toolkits/Frameworks
  – Java, PERL, COBOL, Python, C++, Haskell, …
  – HTML, XML, XHTML, Windows, X Windows, …
  – J2EE, .net, CORBA, CICS, Apache, Tomcat, IIS, …
• UI Layout & Structure
• UI Mockups
• UI Prototypes
• Focus Groups
Design

• **Architecture** *(Tiered/Layered Architecture)*
  – Presentation Logic – “Visible”
    Interactions of UI
    • Look, Feel, Usability patterns for design
    • Technology of the UI
  – Business Logic – rules of application. Its business related behaviors
  – Service (Data Management) Logic – storage, retrieval, persistence, communication
Design

• Techniques
  – JAD (Joint Application Dev) / CDS (Collaborative Design Session)
    • Team works together with users to sketch out possible designs on the whiteboard
  – Information Architecture
    • Taxonomy / Object Hierarchy
    • Task / Feature Prioritization
  – Mockups / Prototypes
    • Low Fidelity – Paper drawings
    • Medium Fidelity – Wireframe Application
    • High Fidelity – Sample application UI
Implementation
“Implementing The Proposed Solution”

• Building the end product deliverable
• Coding Functionality
  – Writing Code
  – “Painting” screens
  – Creating Navigation
  – Creating Graphics
  – Creating Content
  – Creating Data / gathering data
• Fixing bugs you catch
• Usability Lab
  – Watching Users Use Early Versions of the Software
• Beta Testing
  – Prerelease Candidates Used By Users to get feedback
Testing & Fixing
“Verification and Validation of the Solution”

• Functionality
  – Does the action request result in the result that should happen?
• Usability
  – Have Users Use the Software and Watch Them, Solicit Feedback
• Stress & Load
  – Can it Handle the Strain
• Disaster Recovery
  – How well does the software survive things like network, server, disk, database, … crashes?
• Security/Penetration Testing
  – is it secure? Can it be hacked?
• Alpha Testing
  – Early in Implementation, place the software in the field in the hands of real users knowing its mostly not implemented and working just yet
• Beta Testing
  – Late in Implementation, place the software in the field in the hands of real users knowing its not completely working, but should be very close to release
Testing

• Types
  – Unit
    • Do Individual Functions Do What They Should?
  – Integration
    • Does All The Code Work When Combined Together?
  – Usability Testing
    • Navigation
    • Ease of use
    • Aesthetics – is it pretty?
    • Content / functionality
  – User Acceptance Testing (UAT)
    • The Users Test It and Agree It Does What Was Promised
  – System Deployment Testing
    • Load, Stress, Disaster, …
Documentation

• Online and/or paper
  – HTML, PDF, Word, help format, txt
• Developer Docs
  – System Architecture Documentation
• User Docs
• User Tutorials
• User Help documentation
• User Helpdesk documentation
  – How To
  – Known Bugs & Workarounds
• User Documentation
• User Tutorials
• Security Documentation
• System Administration docs
  – Installation, Runtime, Maintenance, Errors, …
Training
“Learning to Use the Solution”

• User
• Helpdesk
• Sys Admin
• Trainer Training
• Developer Training (to add new features later)
Packaging

“Packaging Solution for Distribution and Installation”

• Installation Scripting & Testing Installation Scripts
• Bundling – Zip/Jar/War/…
Deployment

“Putting the solution in the Hands of The Users in the Field”

• Installation
• Monitoring Software in Production
• Preventing Problems
• Remediating Problems
• Keeping it Running
Managing the Project –

• Planning
  – Figuring out what can be done (deliverables), by when, by whom, for how long, and how will we know when it's completed
  – What is needed to complete this project?

• Facilitation
  – Getting Tools, Machines, People, Information, … for the project
  – Managing Meetings

• Coordination
  – Scheduling between Users, Customers, Sponsors and Team Members

• Oversight & Coordination Oversight
  – Reviewing Deliverables and Progress
  – Tracking What is Done and What's Left to Do
  – Insuring that things get done, get documented and meet quality requirements

• Resource Management
  – Managing the People and Helping Them to be Productive
Design Process – In Depth

Information Gathering and Analysis Disciplines
Information Gathering

“Gathering information about the problem and constraints of the solution”

- Gathering information and specific requirements
- Techniques
  - Focus Groups of Users
  - Individual Interviews with Users
  - Observations of Users Doing Their Jobs
  - Workflow, Task and Business Process Review
  - Reviewing Existing Systems
  - Document review and business knowledge research
  - Use Cases & Scenarios
    - Describe, in Text How a User Would Use a System to Accomplish a Goal
Information Gathering - Observation

• Watching users doing what they do
• Looking for general classes of behaviors / actions based on seeing many specific behaviors. Ex:
  – Duplicating a contract, duplicating a report, duplicating a brochure
  – A general class of behaviors called “Copying”
• Understanding the steps taken to achieve a goal.
• Understanding those goals.
Information Gathering - Interviews

• Using the observations to discuss with the user.
• Getting a better understanding of the goals, steps and terminology of the user.
• Documenting the users relative frequency of users actions
• Understand the general characteristics of users including age, education, expertise, computer familiarity, training, …
Information Gathering – Existing Systems

• Examining the functionality present in existing software and devices used by the user currently.

• Helps to add information about what features, functions, tasks, goals, sophistication the user may already expect be supported by any new system.
Information Gathering – Doc review and business knowledge research

- Examining the literature about the business domain
- Gather general business processes, terminology, rules, roles, ...
- Adds more information to the analysis
- See how others maybe doing it differently that these user’s have
Analysis - User Analysis

- User Goals
- User Types
- User Tasks
- User Scenarios / Use Cases
Analysis - User Analysis

- User Goals definition
  - What is it that motivates the users to do the business process?
  - If an done on behalf of another, what is the indirect user’s goals?
  - Find abstract classes of goals. Ex: Make Money is a general class of goal. EarnInterestFromSavings, EarnMoneyByInvestingInTheStockMarket, GetAPaycheck are specific examples of that general goal.
  - Define the users that share that goal.
Analysis - User Analysis (2)

• User Types
  – Determine abstract classes of users from specific users and their shared characteristics.
  – Define those characteristics.
  – Ex: Mike, Sheng, Sam, Cindy, Logan and Oleg are present in class. Except for Logan and Oleg, the rest are here to learn the material presented in the class and they pay to attend the class. They are all Students. Oleg is here to help teach the class and is getting paid to do this. Logan is their to teach the class and gets paid to do so. Oleg is a TA. Students and TAs are abstract classes of users from this group of concrete users.
  – Types are often assignable based on shared goals, characteristics, abilities, etc.
  – Types are also often equivalent to title within the business.
Analysis - User Analysis (3)

• Task Analysis
  – Determine abstract classes of tasks that are done by the users.
  – Define those characteristics.
  – Define the relationships between them.
  – Define the context that they are executed.
  – Define the business process that involve doing a task.
  – Ex: Students do the following things:
    • Sit in a seat, take out a notebook, listen to the professor (intently), ask questions – what tasks and process is this?
    • Review the material in course, sit in a seat, take a copy of a handed out group of papers, read the papers and answer the questions, return the booklet, leave. – what is this?
    • These are part of a larger business process – what is that?
  – Define the business processes and what tasks are used by/shared by those business processes (BP or Biz Proc)
Analysis - User Analysis (4)

• Use Case (UC) Analysis
  – Describe how the user (Actor) would use the system under design (SUD).
  – State the conditions that must be to begin the UC
  – Describe the Goal for the UC
  – Describe the steps that the user would do.
  – Do NOT describe what happens inside the SUD.
Analysis - Use Case Example

- UC0001  - GetMoneyFromATM
- Preconditions: Have account, card, pin
- Scenario:
  1. The Use Case Begins With (TUCBW) the user inserting their card in the SUD.
  2. The SUD displays a screen and asks for a PIN.
  3. The users types in their PIN
  4. The SUD authenticates the users PIN. If invalid display “invalid PIN” and terminate the scenario else continue.
  5. The SUD displays a list of accounts and operations: GetMoney, DepositMoney, CheckBalance and Exit.
  6. The user selects GetMoney.
  7. The SUD prompts for the account and amount.
  8. The user enters the acct and amount.
  9. The SUD debits the account and ejects money.
  10. The user takes the money.
  11. The SUD asks if they’d like to do another operation.
  12. The users selects “No”.
  13. The SUD exits and presents the welcome screen.
- Postconditions: the account is debits properly.
Analysis - Glossary

• Create a list of terms that are relevant to the business domain
  – **Terms** – specific terminology used in this business domain. Ex: DDA, IRA, Trade, Equity, Swap, Market maker, ATM, …
  – **Stakeholders/Users** – can be direct users (employees) or indirect users (customers). Ex: Teller, Broker,
  – **Actors** – a class of user of a system. Ex: Teller, Customer, Administrator
  – **Roles** – a set of behaviors that an Actor may engage in based on certain goal(s). Ex: Head Teller, Junior Teller, Manager, High Value Individual, …
  – Business **goals** – the business benefit that will be derived from a given business process. The reason you request a business process is executed. Ex: Pay For House, Earn Money, Secure My Savings, …
Analysis – Glossary (2)

- Business **processes** – a set of tasks/activities that accomplish a **goal**. Usually something someone would pay for as a unit i.e. you can put a price for that process to be done. An actor playing a role will initiate execution of a business process. Usually stated as a VerbNoun phrase or as a verb phrase. Ex: GetCashiersCheck, MakeDeposit, …
- Business **objects / entities** – high level physical and conceptual things that are manipulated or interact in the conducting of a business process. Business functions/operations – high level operations. Ex: Deposit, DepositSlip, TellerWorkstation, SavingsAccount,…
- Business **rules/calculations** – rules that are specific to this business. Ex: calculating the vacation days for you business. CalculateInterestOnAccount.
  - Usually a sentence or two
  - Including synonyms and related terms
  - Created in parallel with other analysis techniques.
Analysis - Glossary Example

- **Professor** – a person who has knowledge of the subject being taught. Generally slightly eccentric. Syn: instructor, lecturer
- **Student** – a person who attends lecture, listens intently, reads all assigned readings, studies extremely hard and does all homework faithfully.
- **TA (Teaching Assistant)** – helps the professor to teach the class by assisting students, grading homework and proctoring tests. Usually look overworked and underpaid.
- **Homework** – a deviously created set of tasks designed to insure that a student learns the critically important lessons of a course. Often seen by students as a pointless waste of precious sleeping time.
- **GradeHomework** – a business process. Steps are: AssignHomework, DoHomework, SubmitHomework, GradeHomework, ReturnHomeworkToSubmitter.
- **SubmitHomework** – business activity (step) part of a business process (GRadeHomework).
- **TakeNotes** – a business process
- **Notes** – a business object
Review
Process Phases

• Inception (I)
  – Goals
  – define the general business case for the project
  – Define the stakeholders
  – Get a vague idea of the problem being solved (opportunity) and the solution possibilities

• Elaboration (E)
  – Building a prototype of the application – more structural than pretty

• Construction (C)
  – Building the parts of the application, testing them, fixing the bugs.

• Transition (T)
  – Getting the application ready to run in production.
  – Packaging, documentation, training, …

• Production (P)
  – Real users using it and in the process needing support, finding hidden bugs and requesting new features
Parallel Workflows (Disciplines)

- Information Gathering
  - “Gathering information about the problem and constraints of the solution”
- Analysis
  - “Problem Definition”
- Design
  - “Solution Creation”
- Implementation
  - “Implementing The Proposed Solution”
- Testing & Fixing
  - “Verification and Validation of the Solution”
- Training
  - “Learning to Use the Solution”
- Documentation
  - “Creating Manuals and Descriptions of the Solution Implemented”
- Packaging
  - “Packaging Solution for Distribution and Installation”
- Managing the Project
  - “Coordination, Oversight, Facilitation, Tracking and Resource Management”
- Deployment
  - “Putting the solution in the Hands of The Users in the Field”
Focus for the Course

Disciplines:
- Information Gathering
- Analysis
- Design
- Implementation
- Testing & Fixing
- Training
- Documentation
- Packaging
- Managing the project

Phases:
- Inception
- Elaboration
- Construction
- Transition
- Production

As Time Progresses
Information Gathering

“Gathering information about the problem and constraints of the solution”

- Gathering information and specific requirements
- Techniques
  - Focus Groups of Users
  - Individual Interviews with Users
  - Observations of Users Doing Their Jobs
  - Workflow, Task and Business Process Review
  - Reviewing Existing Systems
  - Document review and business knowledge research
  - Use Cases & Scenarios
    - Describe, in Text How a User Would Use a System to Accomplish a Goal
Analysis - User Analysis

- User Goals
- User Types
- User Tasks
- User Scenarios / Use Cases
Homework Project Tracks
Homework #1

• Project Teams and Tracks
• You will be assigned to a group
• The group must decide what project track to pursue based on interests and experience that may help them to define the problem to be solved.
• See the web
  – http://cs.nyu.edu/courses/spring05/G22.2280-001/homework_1.htm
• Due Feb 16, 2004
Project Tracks

• Each team topic is a vague statement about a certain type of solution. Part of your teams project is to take that statement and flesh it out into a vision with specific scope and goals. I have intentionally made these vague to allow your creativity and vision to shine through. You should think about what the future will hold instead of what the presently exists.
## HW1 Artifacts

<table>
<thead>
<tr>
<th>Artifact (Document)</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision Doc</td>
<td>10</td>
</tr>
<tr>
<td>Interviews</td>
<td>10</td>
</tr>
<tr>
<td>Candidate Use Case</td>
<td>3</td>
</tr>
<tr>
<td>Glossary</td>
<td>5</td>
</tr>
<tr>
<td>Actor Definitions</td>
<td>2</td>
</tr>
<tr>
<td><strong>Use Cases (individual)</strong></td>
<td><strong>30</strong></td>
</tr>
</tbody>
</table>

---

Per student 60
Project Tracks
(all ideas copyright, trademarked or patent pending, Poelman & Associates, Inc. unless otherwise copyright, trademark or patented. All rights reserved.)

• **Spacelines Res** - trips to mars? To a distant galaxy? Booking a helicopter? Personal charter hovercraft? Mule? Gondola? Rikshaw? Cab in the city (have it find you)?

• **Autonavigation** - get into the vehicle and it drives you there, how would that work?

• **Electronic Voting** - how about using a kiosk on the street? How about a direct democracy vs. representative democracy? Vote by PDA, Phone, PC, TV, Mail, Email, … How about instantly knowing what the vote total is? How about getting to directly vote on issues instead of having representative government?

• **eworkflow** – email on steroids. Wouldn’t it be great if you could map out the flow of a message from one person to the next to do things like creating, adding graphics, proofing, editing, focus group testing a document? What if you could send out a survey to a group of people and have the survey return itself if not answered within a given amount of time? What if it worked just like internet email does for mail messages?
Electronic Government – design an artificial intelligence local government for your town. You can create and then elect the candidates for mayor, treasurer, chief of police, fire chief, registrar of deeds, ... Then your town would be managed/governed by these intelligent cybernetic life forms. What characteristics would you want to be able to choose from – experience, expertise, diplomacy, bluntness, impulsive, contemplative, lous, brash, polite, tall, short, fat, skinny, blond, brunette, ....?

Future Web - The WWM - instead of the web, what should we have? Death to the browser, but what would replace it? Should it have links? Should the windows be square? Should it be 3D, 4D? Should it have a search engine or should sites collaborate to facilitate your search by sharing information with each other about their content, what types of content should be the standard? Should sites be motion screens instead of flat static pages?

Subway of the Future - no metro cards or tokens? An IP network like system of small subway cars that move like packets through a matrixed network of stations, in all three dimensions, like a really elaborate elevator system that goes not just up and down but between buildings!
Project Tracks

(all ideas copyright, trademarked or patent pending, Poelman & Associates, Inc. unless otherwise copyrighted, trademarked or patented. All rights reserved.)

- **Robot Housekeeper** – what would you need if you had a robot at home to do the house work? How would you “program” it to do the things you wanted it to do? Daily, weekly, Monthly, Yearly Tasks? What features should it do? Not do?

- **Choose your child** – a genetic trait selection system. Pick the features you want, we calculate the cost, you order your new baby and 9 months later you get a FedEx with a baby in it.

- **Live Forever Virtually** – create an avatar that is you, your story and experience that will live on after you are dead. People could come and interact with it/talk with it. Maybe it could learn and act after you are gone based on watching you during your life. Maybe it could listen to your phone calls, emails, IMs, watch you via web cam and learn to mimic you. Talk with your friends and enemies. Talk with you to understand your personality and how it evolves, your preferences, likes, dislikes, strengths, weaknesses, goals, dreams, frears,… Then after you die, it would continue on in that vein.
Project Tracks

- **Cashless Society** - no paper, no credit cards, maybe iris recognition, or DNA?
- **Virtual University** - take classes completely remotely?
- **Cashier-less Checkout** - No scanning, use RFID tags, what would that look like? What about returns?
- **Subway of the Future** - no metro cards or tokens? An IP network like system of small subway cars that move like packets through a matrixed network of stations, in all three dimensions, like a really elaborate elevator system that goes not just up and down but between buildings!