the gamedesigninitiative at cornell university

Lecture 1:

Course Overview; Development Process

CS/INFO 3152: Game Design

- Single semester long game project
  - Interdisciplinary teams of 6 people
  - Design is entirely up to you
- First 3-4 weeks are spent preparing
  - **Labs** to develop basic game concepts
  - **Design** activities to solidify your ideas
  - **Group** activities to help you collaborate
- Remainder of class spent on project

CS/INFO 3152: Game Design

- We provide a basic **milestone** schedule
  - **Today's focus**: the development process
  - Deliverables every two weeks (after week 4)
  - Details on course website:

    http://www.cs.cornell.edu/courses/cs3152
- Games demonstrated at **Showcase**
  - Like BOOM, open to the public
  - Public reaction is part of your grade
  - Submissions posted on the GDIAC website

Course Structure

- **Lectures**: Mondays, Wednesdays, Fridays
  - Of general design and development interest
  - Will include group activities to use the new room
  - Lecture notes posted on website (but incomplete!)
- **Communication Labs**: Tuesdays (usually)
  - Create documents and presentations
  - Satisfies the technical writing requirement
  - See schedule for exact dates

Course Structure

- **Game Labs**: First four Thursdays
  - Special labs for programming or design
  - Complete according to your project role
  - Only INFO has a choice; CS is programming only
- **Playtesting**: Tuesdays for major milestones
  - Submit a **playable** prototype every two weeks
  - Others will playtest your prototype in class
  - We will critique each other's games

This course is a lot of work!

- Expect at least **10 hours/week** outside of class
  - Once the project "starts" in four weeks
  - Typically bare minimum to finish game
  - But if you do this, guaranteed at least a B

Includes

- Time working on game
- Time writing documents
- Time meeting with group

Does Not Include

- 5 days/week in class
- Time spent on readings

This course is a lot of work!

- Expect at least **10 hours/week** outside of class
  - Once the project "starts" in four weeks
  - Typically bare minimum to finish game
  - But if you do this, guaranteed at least a B

If this is a problem, let us know immediately

- Time working on game
- Time writing documents
- Time meeting with group
- 5 days/week in class
- Time spent on readings

Project Groups

- This is a **group-oriented** course
  - 5-6 person teams of diverse talents
  - At least one ≥ 3110 programmer
  - One character designer/artist
  - One user interface specialist
- Groups have been assigned by the staff
  - Taking your preferences into account
  - Groups **must** be in the same section

Game Requirements

- Must be unique with innovative **gameplay**
  - Avoid standard point & click adventures
  - But can take elements from other games
    - **Example**: platformer + something new
- Must be feasible in a semester
  - Avoid full-blown RPGs or real-time-strategy games
  - But can have basic elements of these games
- Must have a single player mode

Game Requirements

- Must develop in the game in **LibGDX**
  - Java-based cross platform engine
  - Has become very advanced, surpassing XNA
  - Can use any IDE, but only IntelliJ is supported
- Must develop a game for a **desktop PC/Mac**
  - Designing gameplay for mobile input is hard!
  - Subject of the advanced class, 4152
- See website for help and resources

Intellectual Property

- Your **group** retains all ownership
  - You can commercialize it later
  - You can make derivative works
  - Individual ownership is your responsibility
- But Cornell gets a non-exclusive license
  - Non-commercial use of final version submitted
  - We can post this version on our website
  - We claim no other rights to your game

Grading Policy

- Mixture of **group** and **individual** grades
- Group grades are same for all group members
  - Group game grade (25%)
  - Technical writing (20%)
  - Class presentations (5%)
- Individual grades distinguish group members
  - Individual game grade (25%)
  - Game Labs (20%)
  - Attendance (at demos) (5%)

Game Grade

- Group grade reflects the game quality

Grade	Criteria
A	Bug-free, Fun-to-play
B	Complete and playable
C	Complete but unplayable
D/F	Serious delinquencies

- Individual grade represents contribution

Grade	Criteria
> Group	Visionary, group MVP
= Group	Good attitude, hard worker
< Group	Produce negative work
D/F	Abandon the group

ENGRC Grading

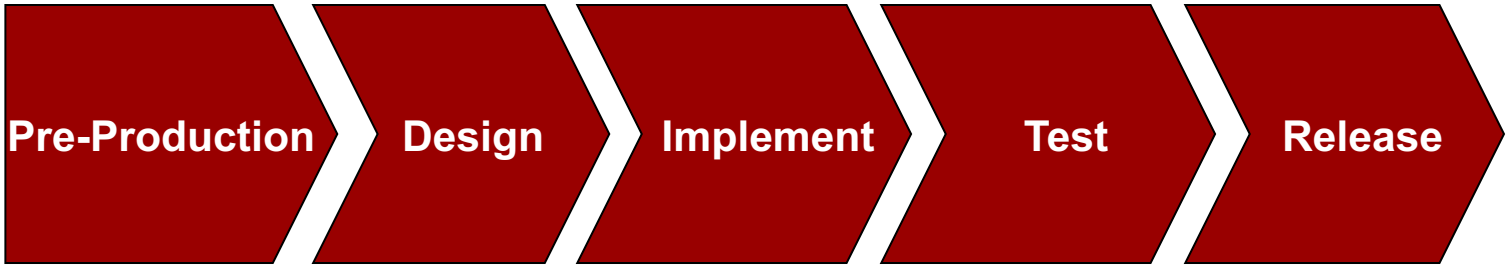
- You **must** enroll in ENGRC 3152 as well
  - No extra work; just what you do in discussion
  - New requirement by school of engineering
- All CS/INFO grades except the game and labs
  - Charter & Group Reports (15%)
  - Course Documents (75%)
  - Attendance & Presentations (10%)
- Typically higher than course grade

Software Development

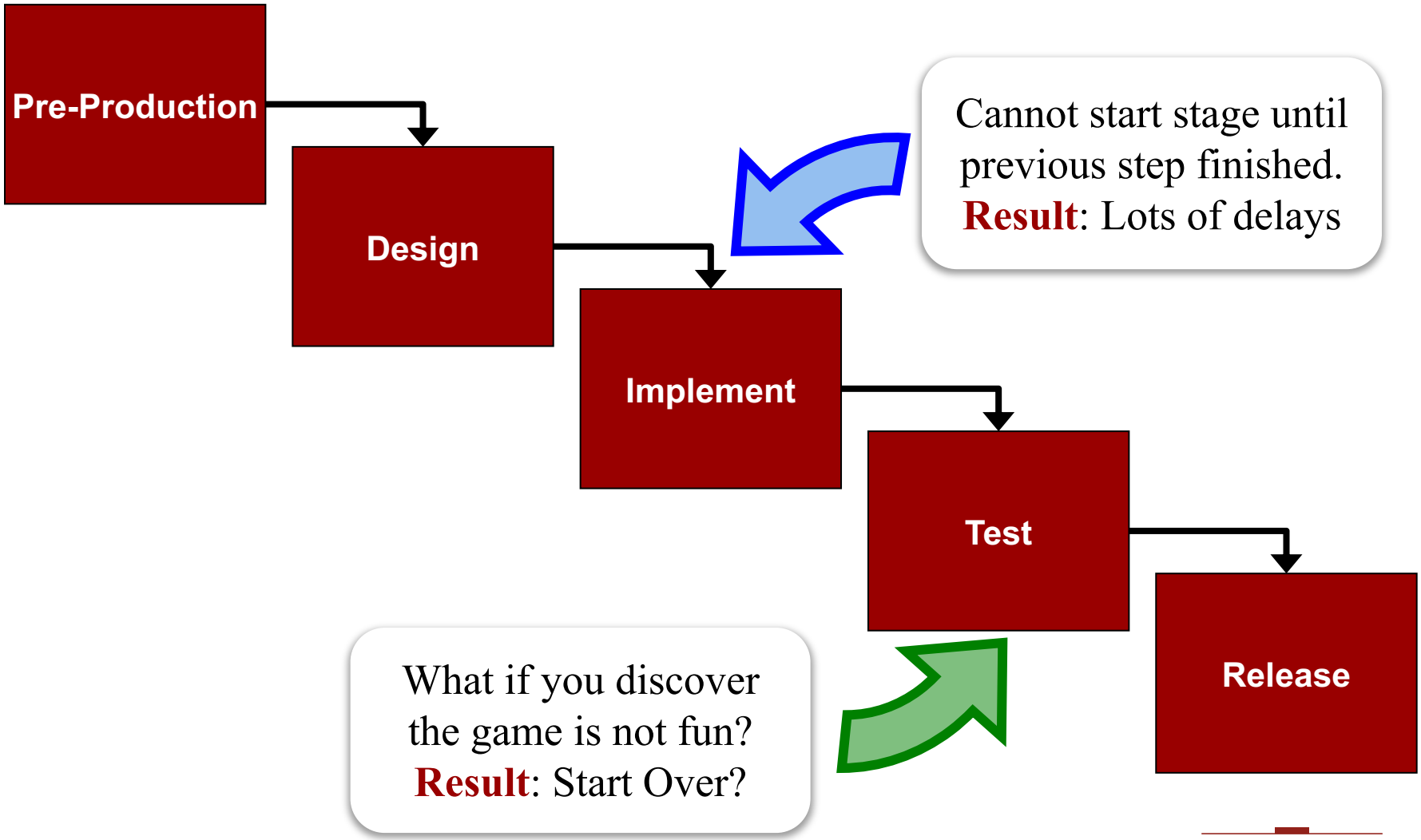
- **Design** process
  - Decide what game you want to make
  - Create a *specification* of your design
- **Development** process
  - Implement your specification
  - Test result to make sure it works
- **Release** (yeah!)

The Traditional Model

- Document extensively; design to specification
  - Design and documents done before coding starts
  - Development follows a specified project timeline
- A general software engineering model
  - Often called the *waterfall* model

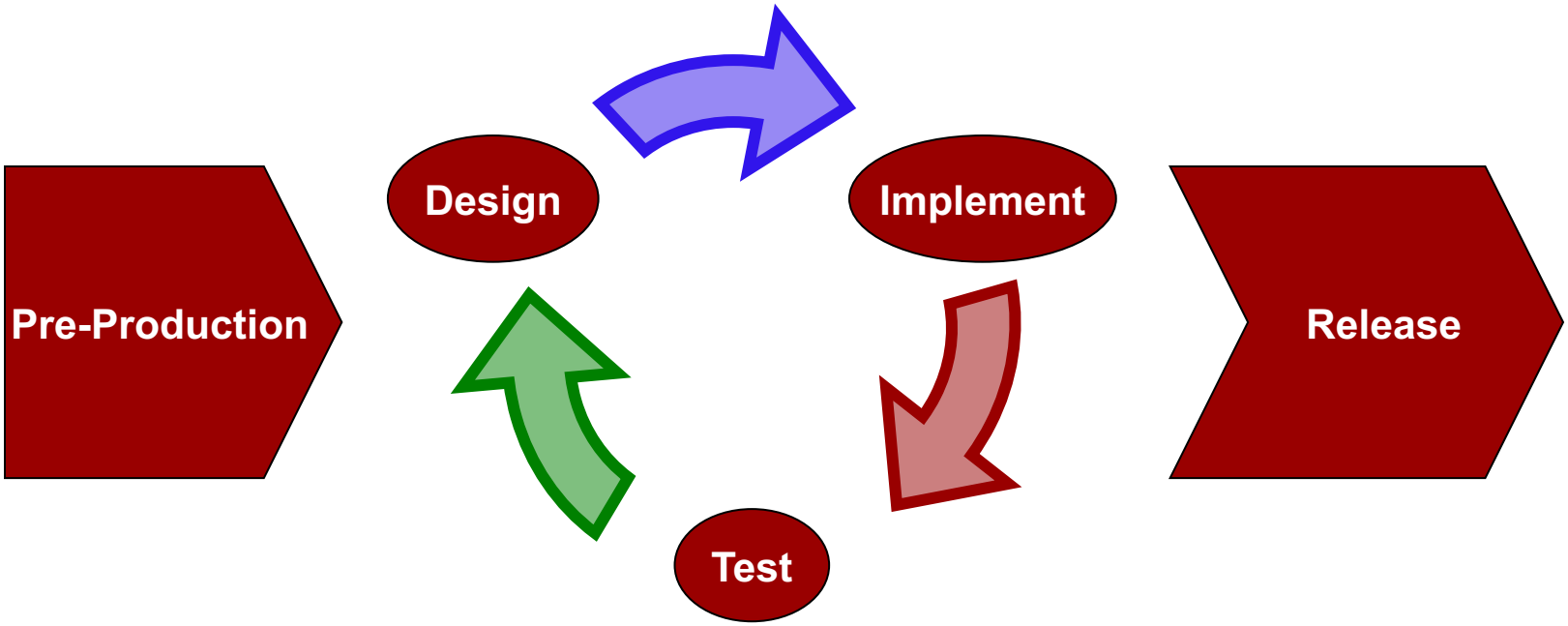


Waterfall Model



The Iterative Model

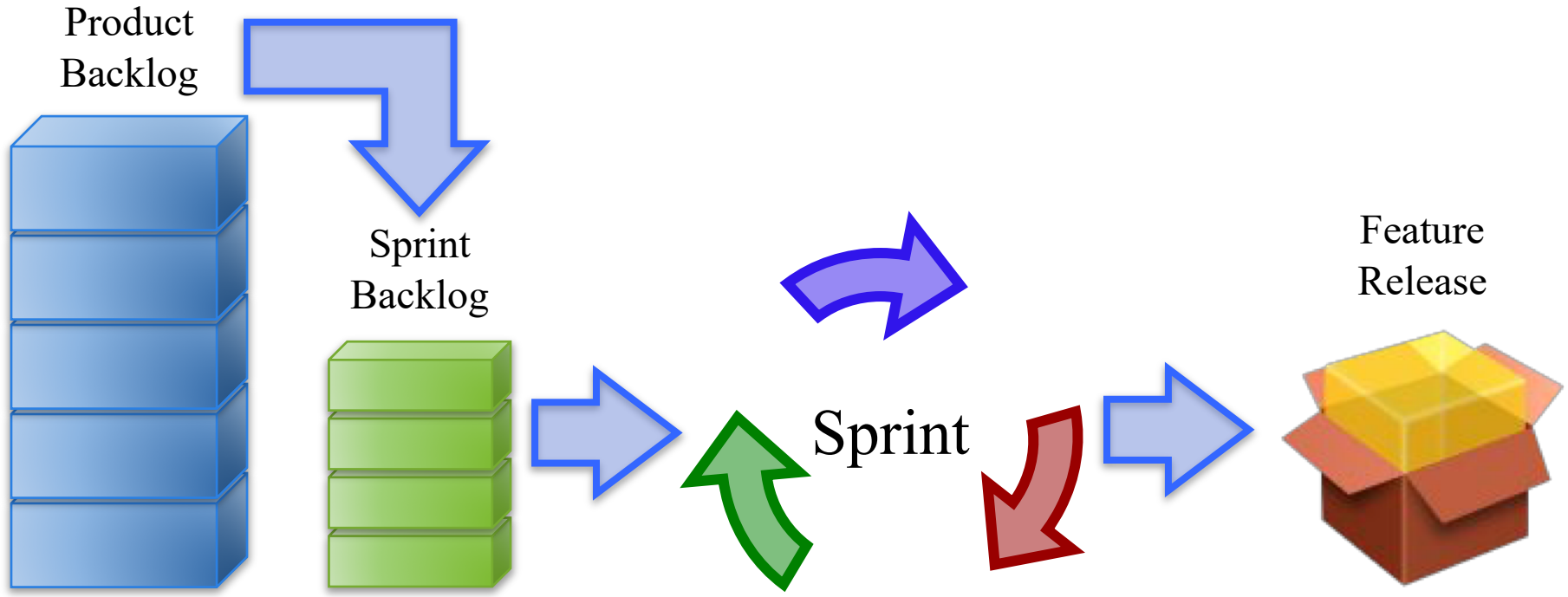
- Cannot evaluate game without playing it first
  - **Iterate**: Rethink design from intermediate results
- Should be playing 20% into development!
  - This requires *prototypes* (may be nondigital)



SCRUM & Agile Development

- Iterative model is called **agile development**
  - The most popular agile method is **SCRUM**
- Key (but not only) idea: **SCRUM sprint**
  - Focus on a small, but testable deliverable
  - 3-4 weeks in industry; 2 weeks in this class
- **Sprint Backlog**: features left to implement
  - Chosen to implement for this sprint
  - Re-evaluate features at end of every sprint

SCRUM Sprint



Milestones

- *Suggestions* for your sprint backlog
  - Flexible enough to handle set-backs
  - Can renegotiate if you get seriously behind

Week 5	**Nondigital Prototype**	2/20
Week 7	**Gameplay Prototype**	3/4
Week 9	**Technical Prototype**	3/18
Week 11	**Alpha (Code Complete)**	4/8
Week 13	**Beta (Feature Complete)**	4/22
Week 15	**Release (Balanced and Tested)**	5/6
Week 16	GDIAC Showcase	5/17

Documentation

- Major part of the development process
  - Why course counts for technical writing
  - Ensures group is always on "same page"
- At every point of development
  - **Pre-production**: concept document, gameplay
  - **Sprints**: reports, architectural specification
  - **Release**: game manual, post-mortem
- Challenge is understanding your *audience*

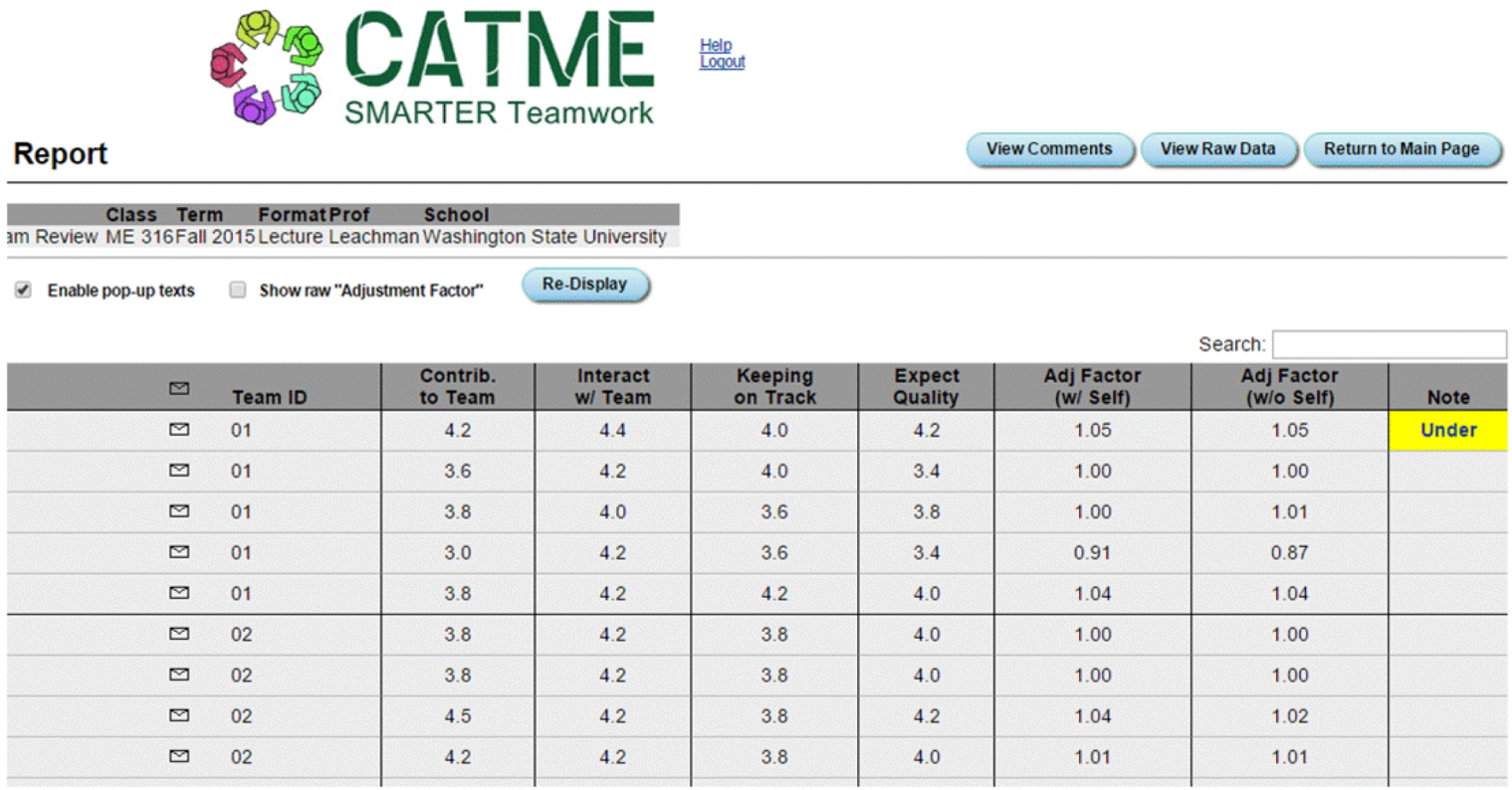
Pre-Production Documentation

- **Concept Document**
  - Describes the basic idea behind your game
  - Communicate core vision without too many details
  - Focus of Communication Lab next week
  - **Audience**: a game publisher (to get funding)
- **Gameplay Specification**
  - Thorough overview of your gameplay
  - Include formal design elements shown in class
  - May change as part of your sprints!
  - **Audience**: new team members (hired later)

Sprint Documentation

- **Reports (every 2 weeks)**
  - Outlines the upcoming sprint (who does what)
  - Reflects on previous sprint (did you meet goals?)
  - Written as a ***group document***
  - **Audience**: your game producer
- **CATME Reports**
  - Online tool for reporting your progress
  - Completed as ***individuals***
  - Allows us to look for problems in group

Using CATME for Reports



	Team ID	Contrib. to Team	Interact w/ Team	Keeping on Track	Expect Quality	Adj Factor (w/ Self)	Adj Factor (w/o Self)	Note
✉	01	4.2	4.4	4.0	4.2	1.05	1.05	Under
✉	01	3.6	4.2	4.0	3.4	1.00	1.00	
✉	01	3.8	4.0	3.6	3.8	1.00	1.01	
✉	01	3.0	4.2	3.6	3.4	0.91	0.87	
✉	01	3.8	4.2	4.2	4.0	1.04	1.04	
✉	02	3.8	4.2	3.8	4.0	1.00	1.00	
✉	02	3.8	4.2	3.8	4.0	1.00	1.00	
✉	02	4.5	4.2	3.8	4.2	1.04	1.02	
✉	02	4.2	4.2	3.8	4.0	1.01	1.01	

http://www.catme.org

Detailed Specifications

- **Architecture Specification**
  - Outline of your software organization
  - Used to distribute tasks to programmers
  - **Audience**: team programmers
- **Design Specification**
  - Outline of your design vision
  - Also includes technical details of asset handling
  - Used for to help designers work together
  - **Audience**: team designers

Detailed Specifications

- **Architecture Specification**
  - Outline of your soft
  - Use
  - **Audience**: team programmers

Directed by the Lead Programmer

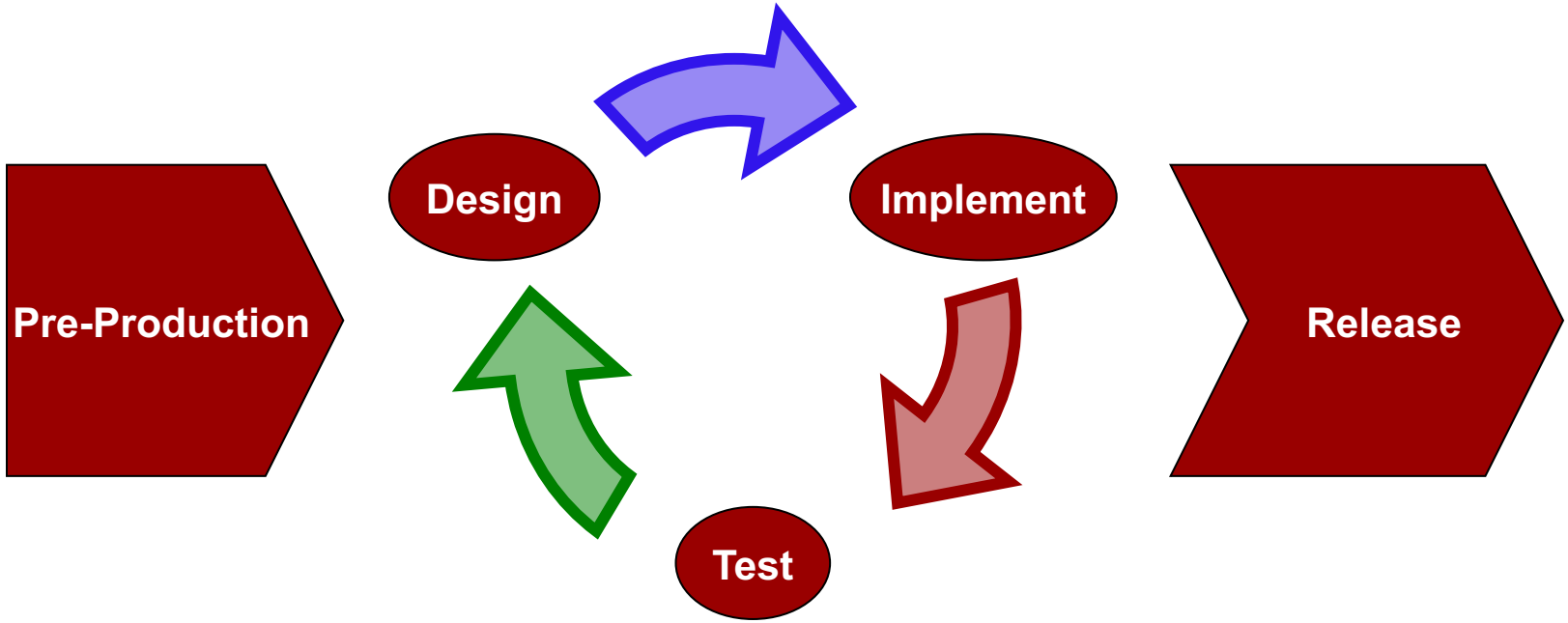
- **Design Specification**
  - Outline of your design vision
  - Also includ
  - Use
  - **Audience**: team designers

Directed by the Lead Designer

Release Documentation

- **Game Manual**
  - Concise description of gameplay
  - Instructions on how to play the game
  - Story, other material to improve the setting
  - **Audience**: your players
- **Postmortem**
  - *Honest* reassessment of what happened
  - What went right; what went wrong
  - **Audience**: yourself (for next time…)

Development Process Review



- **Pre-production**
  - Initial design
  - **Concept Document**
  - **Gameplay Spec**
- **Two-Week Sprints**
  - Playable prototypes
  - **Reports**
  - **Arch/Design Spec**
- **Release**
  - Public Showcase
  - **Game Manual**
  - **Postmortem**

Semester Schedule

Week 1	Team Workflow	1/26
Week 2	Initial Proposal	2/2
Week 3	Revised Proposal	2/9
Week 4	Concept Document (**Project Kickoff**)	2/16
Week 5	**Nondigital Prototype** Milestone Proposals	2/20 2/23
February Break		
Week 6	Gameplay Specification	3/2
Week 7	**Gameplay Prototype**	3/4
Week 8	Detailed Specifications	3/16
Week 9	**Technical Prototype**	3/27

Pre-Production

Development

Semester Schedule

Week 10	Document Revisions	3/30
Spring Break		
Week 11	**Alpha Release (Code Complete)**	4/8
Week 12	Game Manual (Draft based on Alpha)	4/20
Week 13	**Beta Release (Feature Complete)**	4/22
Week 14	Final Portfolio	5/4
Week 15	**Final Presentation (Balanced &Tested)**	5/6
Week 16	GDIAC Showcase	5/17

Development

Release