Project Management and Production

13 Jan 2010
CMPT420 / COMM350
Dr. Sean Ho
Trinity Western University

Quiz1: 5qs, 5pts, 5mins

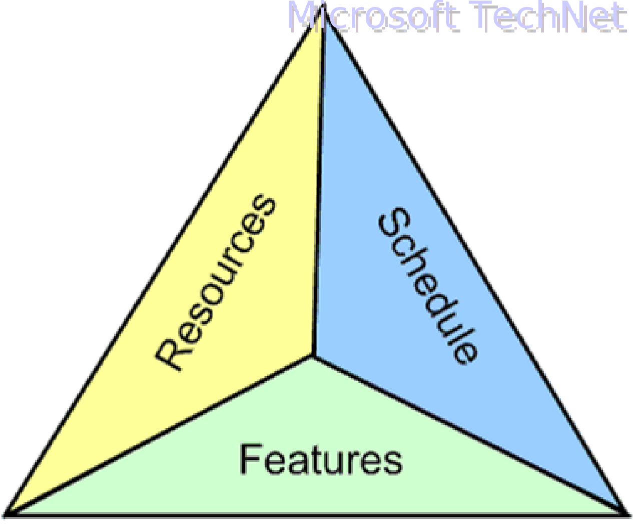
- Name the four stages of a production cycle. (ch1)
- Name two project management methods described in ch3.
- Briefly contrast directed play testing vs. freeform testing. (ch17)
- If a project is getting off track, what are the four fundamental areas to examine? (ch18)
  - *(hint: the triangle diagram in Fig18.1)*
- In ch19, the person who oversees and streamlines the automated build process, working with QA to set a delivery schedule for new builds, is called the ___________.

Quiz1: answers #1-3

- Name the four stages of a production cycle. (ch1)
  - Pre-production, Production, Testing/QA, and Post-production
- Name two project management methods described in ch3.
  - PSP (personal software process), TSP, Scrum, Waterfall, Microsoft Process
- Briefly contrast directed play testing vs. freeform testing. (ch17)
  - Directed play: structured test plan, exhaustive, know what to look for
  - Freeform: jump in and play w/o reading docs

Quiz1: answers #4-5

- If a project is getting off track, what are the four fundamental areas to examine? (ch18)
  - *(hint: the triangle diagram in Fig18.1)*
  - Features, Schedule, Resources, Quality



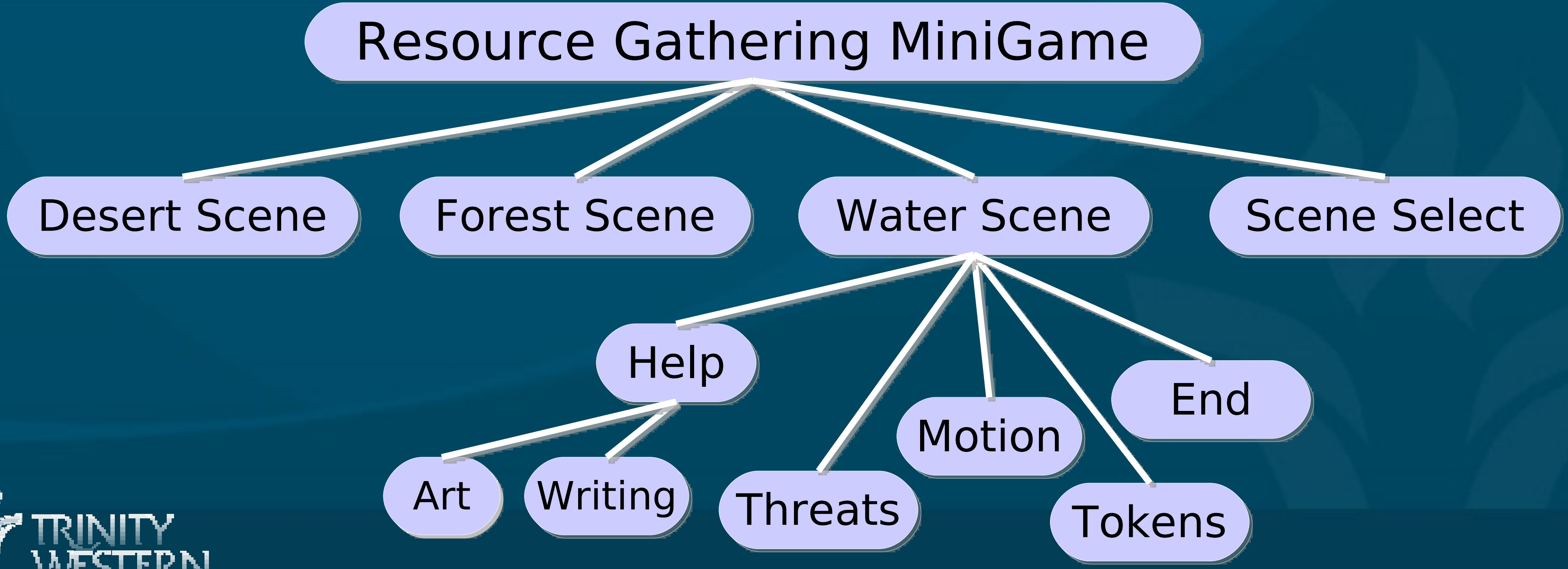
- In ch19, the person who oversees and streamlines the automated build process, working with QA to set a delivery schedule for new builds, is called the ____________.
  - Data Manager (or Configuration Manager)

Outline for Today

- Top-Down vs. Bottom-Up
  - What is a “feature”?
- The Production Cycle
  - Pre-Prod. → Production → QA → Post-Prod.
- Project Management: Scrum
  - Agile vs. Waterfall; Roles; Process
- Project Reviews and Critical Stage Analyses

Top-Down vs. Bottom-Up

- Top-down: break down a complex task into simpler components
- Bottom-up: what tools do we have available, and how can we use them to build what we want
- Tasks (features, assets): bite-size chunks of work, individually designed → produced → tested



Pre-Production

- Concept: mission (of project), concept (of game), genre, stakeholders, risk analysis: → pitch
- Features: decomposition of the task
  - Assets: art, sound, text
  - Software components
  - Prioritize: MoSCoW must / should / could / won't
- Milestones: date + deliverables
- Schedule / Game Plan:
  - Dependencies amongst features/assets/tasks
  - Estimate needed time/resources (this is hard!)
  - Feasibility: reduce features if necessary

Production

- Know what you need to do
  - Communicate expectations clearly
- Track what you're doing
  - Make progress every day!
  - Blog / journal / change log / commit log:
    Which task did you work on? What did you do?
  - Seek help if you're stuck – collaborative environ
- Know when you're done
  - MVP: minimum viable product
  - Learning to let go:
    frequent feedback vs. perfectionism

Quality Assurance

- #1 Rule of QA: It must be by someone else!
  - Checking your own work is good, but not QA
- #2 rule: Get feedback early and often
  - Hallway test: grab someone passing by and show them your work (30-60sec)
  - “Do you like it?” and “Why?”
- Too much feedback can be bad, too
  - Feature-creep and dilution of the orig. concept
  - → QA team manages bug/feature requests, and Product Owner in each team prioritizes
    - More later on Scrum roles

Post-Production

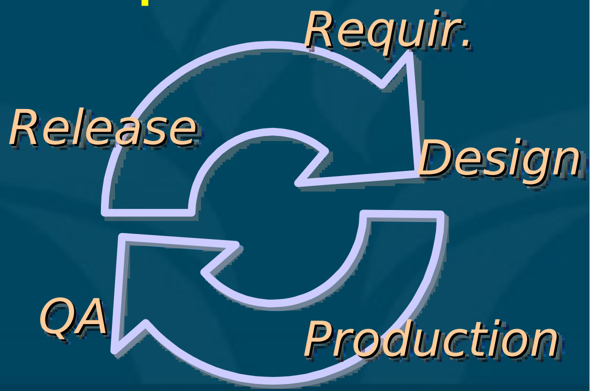
- For the team:
  - Lessons learned, what to improve next time
  - Communication, leadership style, conflict resol, infrastructure, personal strengths/weaknesses
  - Postmortem vs. Critical Stage Analyses
- For the product:
  - Planning releases, versions, future development
  - Copyright and licensing (think about this early!)
  - Archiving documentation, assets, code
  - Point of contact
    - “Nothing disappears on the Web”

Outline for today

- Top-Down vs. Bottom-Up
  - What is a “feature”?
- The Production Cycle
  - Pre-Prod. → Production → QA → Post-Prod.
- Project Management: Scrum
  - Agile vs. Waterfall; Roles; Process
- Project Reviews and Critical Stage Analyses

Waterfall vs. Agile

- "Waterfall" method does each stage completely before moving on, in a rigid fashion:
  Requir. → Planning → Production → QA → Release
  - Hard to determine requirements in advance
  - Hard to estimate needed time/HR/tools
- Agile (aka Spiral) methods are "iterated waterfall"
  - Scrum is one such method
  - Each iteration through the spiral is a sprint
    - Sprints need to be short! 1-4 weeks
  - Early prototypes → early feedback

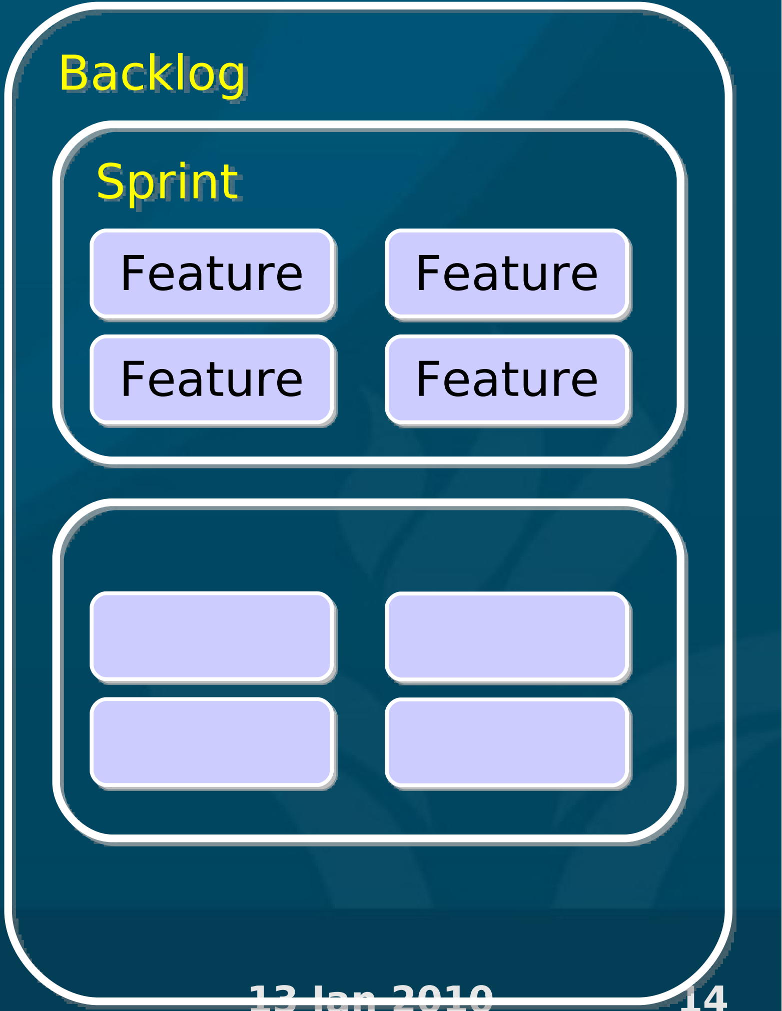


Scrum: Roles

- Small, interdisciplinary / cross-trained teams
  - Everyone knows a bit how to do everyone's job
- “Pig” Roles (committed):
  - Team (5-9 ppl): design, impl., QA, commun., ...
  - ScrumMaster: protect, keep Team on-task
  - Product Owner: “voice” of the client, writes use-cases (“stories”, requirements), gives feedback on results to Team
- “Chicken” Roles (involved):
  - Client, stakeholders: business need, marketing, artistic vision, design studio, etc.

Scrum: Process

- Prioritized features go in backlog
- Divide backlog into sprints (1-4 wks)
  - Burndown chart tracks progress
- Sprint planning meetings
  - Choose features to tackle
- Daily stand-up scrum meetings
  - Time-boxed to 15min
  - Only “pigs” may speak
- Sprint review meetings
  - Get client feedback
  - Team feedback on process



Outline for today

- Top-Down vs. Bottom-Up
  - What is a “feature”?
- The Production Cycle
  - Pre-Prod. → Production → QA → Post-Prod.
- Project Management: Scrum
  - Agile vs. Waterfall; Roles; Process
- Project Reviews and Critical Stage Analyses

Project Reviews

- Post-mortems (done after project completion) vs. Critical Stage Analyses (done after each sprint)
- Retrospective (review the past):
  - Achievements since last time
  - Compare with plan:
    On-task? On-time? Under-budget?
  - Roadblocks: are we waiting on anything?
- Prospective (plan for future):
  - Potential risks
  - Resources needed (time, people, tools)
  - Decide on next features/tasks to work on

Critical Stage Analyses

- CSAs are a way of doing frequent, regular reviews
  - At critical milestones, or even after each sprint
- Team self-analysis: each member prioritizes:
  - 5 things that went right (in this past period)
  - 5 things that went wrong
  - 5 things that could be improved (for next time)
- Team lead compiles and distributes results
- Discuss as a team at project review meeting
  - Stay positive! Focus on change for the better

Communication at scale

- Meetings are about communication:
  - Gather information (status, feedback)
  - Make decisions
- As the time-scale increases, the communication spans a larger and more diverse group of people
  - Seconds: pair / collaborative work on a task
  - Minutes: unit testing, self-check
  - Hours: commit a feature, complete an asset
  - Daily: 15min stand-up Scrums w/team
  - Weekly: acceptance tests, update other teams
  - 1-4 Weeks: sprints, milestones (CSAs), releases

OK, Now What?

- Figure out which stage of the production cycle your team currently is at
  - Planning/sched. → production → QA → release
  - One sprint per mini-game?
- If you're unsure, then you're probably in planning!
  - Break it down into bite-size tasks
  - Dependencies, time estimates → schedule
- Be pro-active in figuring out what you should do, and COMMUNICATE!
  (with teammates, with your team lead,
  across teams, with us, with external testers)