

# Finalling: Releasing and Post-Production

3 March 2011

CMPT420 / COMM350

Sean Ho

Trinity Western University

# Quiz ch23-25

- Name 2 (out of  $\approx 8$ ) checklist items that should be on a code release checklist.
- What 4 questions should a postmortem focus on?
- What document does a postmortem produce?
- What is the purpose of a closing kit?
- What 4 (categories of) items go into a closing kit?

# Quiz ch23-25: answers #1-3

- Name 2 (out of  $\approx 8$ ) checklist items that should be on a code release checklist.
  - Legal/©/EULA, support info, localize, demos, ratings, console/TCR approval, QA approval
- What 4 questions should a postmortem focus on?
  - Achieve goals? Resources/expectations realistic? What went right/wrong? Lessons learned?
- What document does a postmortem produce?
  - Lessons-learned document

# Quiz ch23-25: answers #4-5

- What is the **purpose** of a closing kit?
  - Enable a future dev team to continue development
- What **4** (categories of) **items** go into a closing kit?
  - Assets, tools, code, documentation

# Outline for today: Finalling

## ■ Milestones for code release:

- Alpha → Beta → RC → Gold

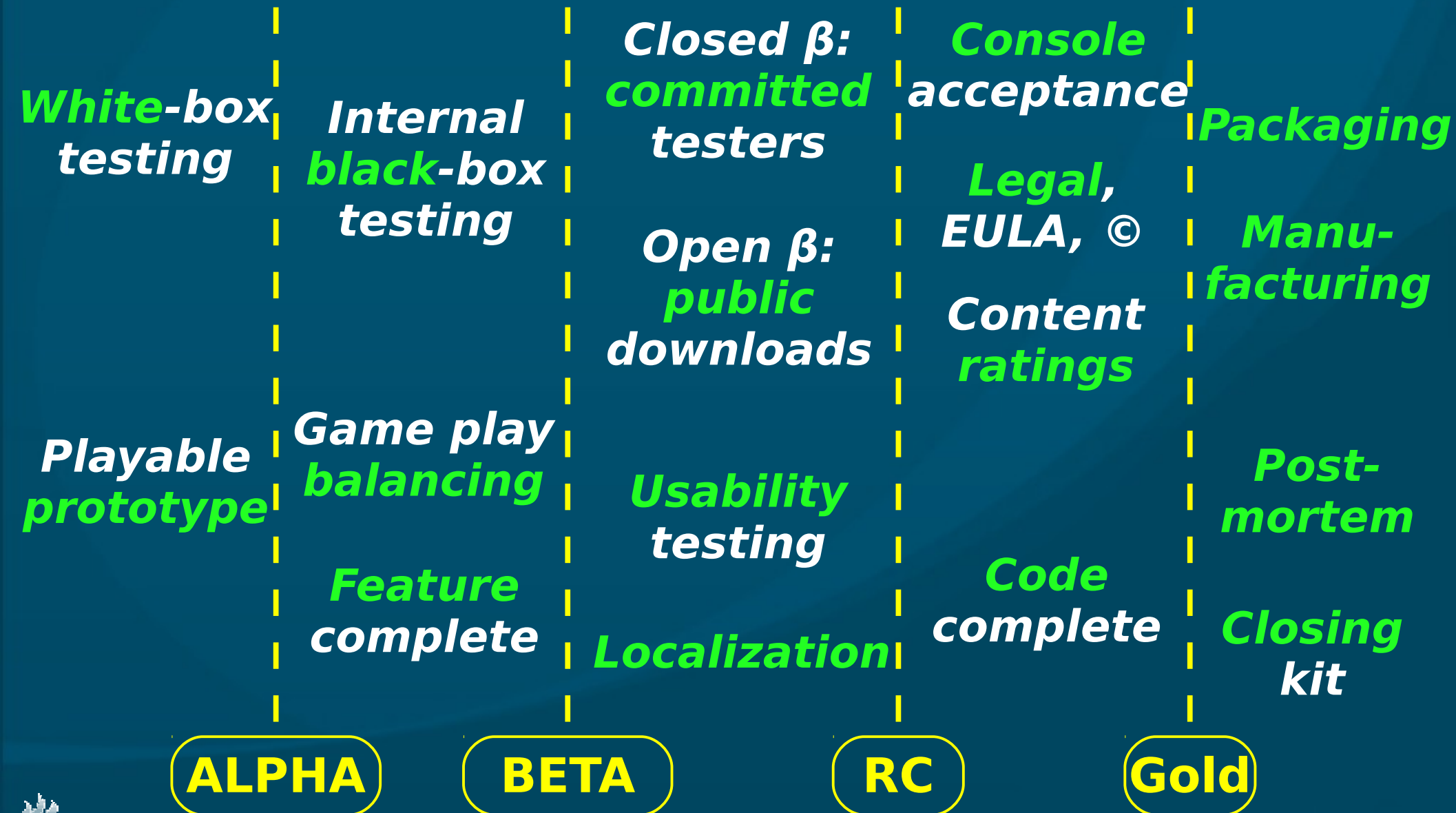
## ■ Closing kits

- Docs + Code + Assets + Tools

## ■ Postmortems

- Orig. Plan → Reality → Root Causes → New Goals

# Milestones for code release



# Closing kits

- Live developer systems (e.g., Trac) may be shutdown after production is complete, so
- Closing kit is an archive of the production
- Everything a new team would need to continue development or produce a sequel:
  - Documentation
  - Source code
  - Assets
  - Tools

# Contents of a closing kit: 1

## ■ Docs:

- Design docs, UI mockups
- QA test plans / checklists
- External libraries (w/ver): PyGame, PyMunk, ...
- Dev tools: Hg, PyDev
- How-to install dev setup, how-to build
- Asset production pipeline

## ■ Code:

- Incl. snapshots of gold master code, local tools



# Contents of a closing kit: 2

- **Assets:** incl. all asset **checklists**
  - Original source for **art** (so can tweak)
  - Uncompressed **audio** (music, effects, VO)
  - All in-game **text** (dialogue, tooltips, labels, help)
  - **Video:** in-game FMV, promo trailers
  - **Localized** versions of everything (+ checklists)
- **Tools:** in-house or 3<sup>rd</sup>-party
  - Dev/**build** tools, **asset** production tools, **localization** tools, special plug-ins

# Postmortems

- Immediately after shipping final product
  - While the experience is still fresh in minds
  - Budget time for it in schedule, before folks leave
- Production process, not details of project
  - Lessons need to be portable to next project
- Cultivating a culture of learning
  - Don't: “Whew, I'm done! Throw it all away!”
  - Do: “What can we learn for next time?”
  - Learn ... history ... or repeat ...

# After Action Reviews (AAR)

- Conceptual forefather of postmortems
- Developed by **U.S. Army**, still used in training, peacekeeping, and combat
- What was **planned**? (**objective**)
- What actually **happened**? (**reality**)
- What were the root **causes**? (**learning**)
- What will we **sustain** or **improve**? (**goals**)
- *Further reading: USAID ppt, USArmy TC25, Signet Consulting*



# Case study: Valve's “Portal”

- (GamaSutra / Game Developer Jan 2008)
- 8 fresh grads from DigiPenIT
  - Senior project: *Narbacular Drop*
  - Job fair → hired by Valve → dev *Portal* in 26mos
- Play tests: e.g., difficulty ramps
- Story: GLaDOS character
- Using frameworks: extending Source
- Performance: LOD+stencils for render speed



# Square/Enix FFXIII

GamaSutra: "What Went Wrong?"  
and "Behind the Scenes of FF XIII"



## ■ Poor communication:

“During the development of Final Fantasy XII, the **pressure to succeed** was at such a high point that we were on the brink of losing control during even the slightest **misunderstanding**. What happened was our team was given the **freedom to make changes** at various stages of development, but the adverse affect of this freedom was **miscommunication, confusion, and disorder**. How work was to be **distributed** was also often ambiguous, which contributed to the problem.”

## ■ Attaining a shared vision through demo:

“With a **tangible** version of the game that could actually be **played**, internal debates transitioned from theoretical discussions based solely on abstract concepts to **concrete dialogue**. The demo not only **unified** the vision and understanding of the game’s direction across the entire development team, but it was also the first time that everyone could **see** exactly how the **assets** they worked on would function within the game”

# Further reading

## ■ Postmortems:

- GamaSutra's extensive archive (GameDev mag)
  - ◆ 2009 summary: top 10 “What Went Wrong?”
- Vector Unit's “*Hydro Thunder Hurricane*” (≤7-man, 14-mo, startup)
- AtomicBoy's “*Empires of Steel*” (1-man failure)
- Lazy8's “*Cogs*” (indie success)