



# Shared Technology at Rare: Good and Bad

- Tom Grove
- GDC 2007 San Francisco
- [www.rareware.com](http://www.rareware.com)



# Outline

- Who are Rare?
- The Shared Technology Group
- Lessons Learnt
- Was it worth it?
- Summary
- Questions?

XBOX 360





## Rare Ltd

- Part of MGS
- Creatively Lead
- Multi Title
  - 2-4 360 teams
  - Prototype teams
  - DS / Handheld Team
- Support Teams
  - Shared Technology Group
  - Audio Department
  - Art Asset Group





# Shared Technology Group

- Background
- Motivation
- Development
  - Initial plan and focus
- Review of initial approach

XBOX 360





# STG: Background

- History
  - “RnD” Setup in 1999; 5-6 inexperienced developers, 1 lead
  - Currently 20 developers, 2 leads and producer
- Used by all console titles since 2000
  - First title: Starfox ( Game Cube )
  - Six major titles so far



# STG: Motivation

- Why was the group setup?
- Reduce Duplication
  - Over five different engines on N64
  - Development cost expected to increase
- Disseminate best practice
  - Best of breed
- Share research
- Support art and design





# STG: Initial Plan

- Interview teams to see what they do
- Develop a shared engine (“r1”)
- Ready for teams moving from N64 to GC
- Game development model

XBOX 360



# STG: Initial Focus

- Response to perceived problems
- Strong focus on art-pipeline
  - Reflection of creativity lead development
  - Respected art tool in previewer
  - Artist authored shaders
- Emphasis on runtime performance
  - Expectations from single platform history

XBOX 360





# Review

- Successes
  - Accurate art tool reflection
  - High runtime performance
- But key weak areas
  - Development Process
  - Distribution and Support
  - Client Relationships
- We examine these next



# Technology Development

Then and Now





# Technology: Then

- Artist directed technology
  - Confused communication
- Focus on “next-gen” features circa 2000
  - High-order surfaces, physics, ...
- Too much emphasis on runtime
  - Single platform culture
  - Naïve content expectations

XBOX 360



# Technology: Then

- Reactive development
  - Polish and optimisation postponed
  - Favours vocal minority
- Too little experience
  - Code quality
  - Focus on “cool” features

XBOX 360



# Technology: Now

- Pro-active Coordination with teams
  - Agile development ( scrum-like )
  - Transparent “ring-fencing “ of capacity
- Producer
- Peer code reviews
- Components based
- Technology is not the hardest part...



# Component Based

- Not building an engine
  - Clients already had engines ..
- Set of independent components
- Allows for middleware
- Clients take suitable components
- Components support customisation
  - Important in getting support of graphics engineers





# Component Catalogue

- Animation
- Rendering
- Art tool support
  - Plug-ins
  - Exporters
- Art-pipeline
  - Max and Maya
- Tools
  - Asset management
  - World building
  - Asset previewer
- Fonts
- Data reflection
- Collision detection
- Maths
- Profiling

XBOX 360



# Component Use

- Kameo: Elements of Power
  - Used all components, but with custom lighting
- Perfect Dark Zero
  - Custom Deferred Renderer built on top of existing pipeline components
  - Havok for physics and collisions
- Viva Pinata
  - Only animation and low-level components
  - Co-existed with an existing renderer





# Distribution and Support

Then and Now





# Distribution and Support: Then

- Did not really consider distribution
- Initially planned quarterly releases
- But taking a new version painful
  - Development cycles out of sync
  - Asset and code build times
- Poor model for team code changes
  - Re-integration of local changes



# Distribution and Support: Now

- We see ourselves as much a service as a product
  - “fire and forget” does not work for middleware
- Improved build quality
- Deprecation policy
- Better source control tools ( source depot )
- Better customisation
- Case officers



# The role of the case officer

- Developer allocated to each game in production
  - Prototypes do not generally need one
- Bridge between the game team and STG
  - Accountable developer
- Has a personal stake in the product
- Responsible for arguing the clients case
  - On-site customer in agile methodologies





# Client Relations

Then and Now





# Client Relations: Then

- Critically important
- STG did fit into the development culture
  - Competitive teams
- Poor feedback between teams and STG
- An Us-vs-Them situation developed

XBOX 360





# Client Relations: Now

- Involve teams in monthly sprint planning
- Quarterly product review meetings
- Game teams mentor STG developers
- Case officers again
- Informal monthly technical lead meetings
  - with biscuits!
- All new starts come through STG
  - Removes the “us-and-them” distinction



# Was it worth it?

- Modest team sizes (  $\approx 30$  ) outside of crunch
- Three titles shipped in last two years
- Game teams less technology focused
- Improved development atmosphere
- Preserved core values
  - Still art / design lead
  - Still have strong team identities





# Future

- Binary changes still a problem
  - Case officers feel the pain
- Documentation
  - Recruitment is difficult
- Tools still need work
- Build times a problem
  - How to balance re-factoring against cost to clients?

# Summary: Lessons Learnt

- Client Relationships
  - Critical to build culture where good will is assumed on both sides
  - Face to face meetings
  - Case officers
- Support and Distribution
  - Software as service
- Development
  - components
  - Agile development



The image shows the interior of a Norman church, likely a nave or choir. It features a series of large, square columns supporting a high, vaulted ceiling. The architecture is characterized by thick walls, rounded arches, and decorative carvings. A large, circular opening (oculus) is visible in the upper left. The lighting is warm and focused on the central area, creating a sense of depth and grandeur.

Questions?

[tgrove@rare.co.uk](mailto:tgrove@rare.co.uk)