“You Might Dig”
Transition Phase Project Status Assessment

1. Did the project meet its initial objectives? What were they?

Yes. The initial objectives were to create a web application which:

- Gathers and stores games and game information from various online stores
- Provides an easy to use interface for users to browse and rate the gathered games
- Allows users to specify the kinds of games they enjoy
- Generates recommended games for users based on their preferences and the similarities they have with other users and the games they’ve rated highly

2. Was the primary stakeholder satisfied with the end result?

Our project adopted a “virtual” stakeholder based on our prospective users and ourselves (the users of the system) and based our stakeholder satisfaction off the feedback given by our testers. Based on their feedback we feel that our *stakeholder* is satisfied with the end result. (Please see this and this form *Note: Most users did not use the follow up form and instead responded to the original form again*)

a. Did they have any comments or further requests?

Our virtual stakeholder did have many comments and further requests - for the most part we accommodated the changes requested. (Please see this sheet for more information on requests and our reasoning)

Some requests of note:

- Resizing the rate game stars
- Reordering the user options menu
- Adding functionality to the “Sign in to rate” button on game pages
- Re-set some genre characteristics based on feedback

3. Did you manage to stay on schedule and complete on time?

During the transition phase - our iterations became somewhat shifted and messily organised however they were completed on time and our project was overall ready for submission on time.

Throughout the entire project our schedule did run into delays as noted in previous phase status assessments (notably construction phase)

a. If not, what issues caused delays?

The most notable *overall* delay that did not impact this current iteration occurred during the construction phase, after returning to ITC309 after the mid year break when team members took some time and prompting by Jim to return to the expected workload.

b. If you charged $50 per programmer hour, how much would your project have cost?

	Documentation	Cost	Programming	Cost	Total	Cost
Inception 1	21	$1,050	0	$0	21	$1,050
Inception 2	24	$1,200	18	$900	42	$2,100
Elaboration 1	8	$400	6	$300	14	$700
Elaboration 2	0	$0	0	$0	0	$0
Elaboration 3	8	$400	16	$800	24	$1,200
Elaboration 4	0	$0	21	$1,050	21	$1,050
Elaboration 5	21	$1,050	5	$250	26	$1,300
Construction 1	0	$0	16	$800	16	$800
Construction 2	2	$100	14	$700	16	$800
Construction 3	11	$550	26	$1,300	37	$1,850
Construction 4	29	$1,450	7	$350	36	$1,800
Construction 5	12	$600	11	$550	24	$1,150
Transition 1	10	$500	3	$150	13	$650
Transition 2	2	$100	1	$50	3	$150
Transition 3	17	$850	0	$0	17	$850
TOTAL	165	$8,250	144	$7,200	292	$15,450

4. Risks and issues encountered and dealing with them

Towards the end of the construction phase, the production environment ran out of space after storing ~20GB of game screenshots and thumbnails. This was remedied by adding an external storage volume to the production VM and storing the screenshots there.

A major risk/issue has been that our system might not gather enough games. Unfortunately this has wound up being the case. While our system is entirely capable of gathering and storing the number of games desired, many of the newer and searched for games aren’t found in our system. This isn’t to say the system doesn’t store many games, there are thousands currently stored on the system and we could have stored thousands more if we constantly had our gatherers running. The reason this risk came to be is that some stores rate limit the number of requests that can be made to their services within a certain time period. For instance steam will rate limit after only a few dozen requests within a minute, and the gatherer system must wait until its rate limit is reset.
This could be remedied (as noted in the issues sheet) by possibly using multiple VMs or a machine with multiple IP addresses to perform the gathering to reduce the effects of rate limiting. This solution was not chosen currently due to having a high cost which seemed impractical for the current scope of our current project. Especially when the desired games *could* be gathered, given enough time. (And then newly released/announced games could be selectively gathered without having to worry about rate limiting as they appear)

Quite a few suggestions and issues arrived during beta testing.

The majority of suggestions made by testers were implemented quickly, and some other minor issues not reported by beta testers were also resolved (See issues #9 and #10 on github) requiring usually simple changes to the system.

Some larger scale issues that would require a larger rework of the system have unfortunately been left unresolved:

- Genres wind up duplicated due to the multiple stores having different ways of wording similar genres e.g. “RPG”, “Role-Playing”, “Role-Playing Games (RPG)”. A solution has been discussed that would involve providing an admin setting to remap genres - such that while gathering, “Role-Playing” would become “RPG”. However the time it would take to fully resolve the issue, it was decided to leave the issue for the time being.

- Some games wind up with multiple items due to being found on multiple stores and there being no merge occurring. While not strictly an issue, the suggestion made a very good point, and it is felt that the game items should be merged considering they are covering the same game (though it could be argued that the game may be rated differently on different platforms) The solution would be to compare a stripped down version of the titles (removing things like © and forcing lowercase for comparison) and adding platform information to existing ones. Once again the scale of issue was deemed too large for the remaining time the team had.

Both the above issues were submitted during the 2nd round of beta testing, closer to project end and as such the amount of time to work on them was much reduced.

5. Did the project methodology work?

Overall - yes. The project was completed and all team members did contribute to that end. Some issues were encountered during the project as a whole but were resolved. (And can be read about in other status assessments, and in the section above)

a. What aspects helped?

A large focus on planning during the inception phase helps get a very solid idea of what the primary goals of the project were and helped to maintain that aim throughout development rather than shoehorning different aspects into the project and adjusting others to fit.

The iterative structure and separate phases more easily allows the evaluation of team member contributions and their involvement in the project as well as allows team members to find tasks to perform more easily.

b. What aspects hindered?

A focus on a tangible shareholder often found us confused especially during evaluation of goals. Once we adopted the “virtual” stakeholder we found the process much smoother.

c. What should be done to improve the process?

Formal code review is an important part of collaborative projects. Something we generally did not perform due to our iterations often having much work being completed and requiring merging and not having the time to perform formal code reviews.

d. What procedures could be useful in future projects?

The amount of thought put into choosing an architecture should be very helpful in future projects, where without proper planning, a project may become nearly fully developed on an architecture not suited to the project itself.

6. Steps to improve the likelihood of project success in the future

a. What changes would speed things up while improving communication?

Set and moderate **strict** weekly meeting times, ensuring team members attend and the present plans, issues and concerns are communicated quickly and ensures other team members do engage with the communications.

Team members more closely paying attention to the work the others are performing

7. What lessons have you learned that will apply in future projects?

- To ensure better team communication, and making sure all team members will hold the others accountable to their work and participating in the project itself - ensuring the responsibility doesn’t fall to a single person.
- **Strict** weekly meetings should be set (not just the oversight meetings) to organise iteration plans and check progress
- Holding team members more directly responsible for contributing to the overall progress of the project is important
- Directly referencing the team charter more often and being more familiar with it is needed to be able to quickly identify when a team member isn’t engaging with the project