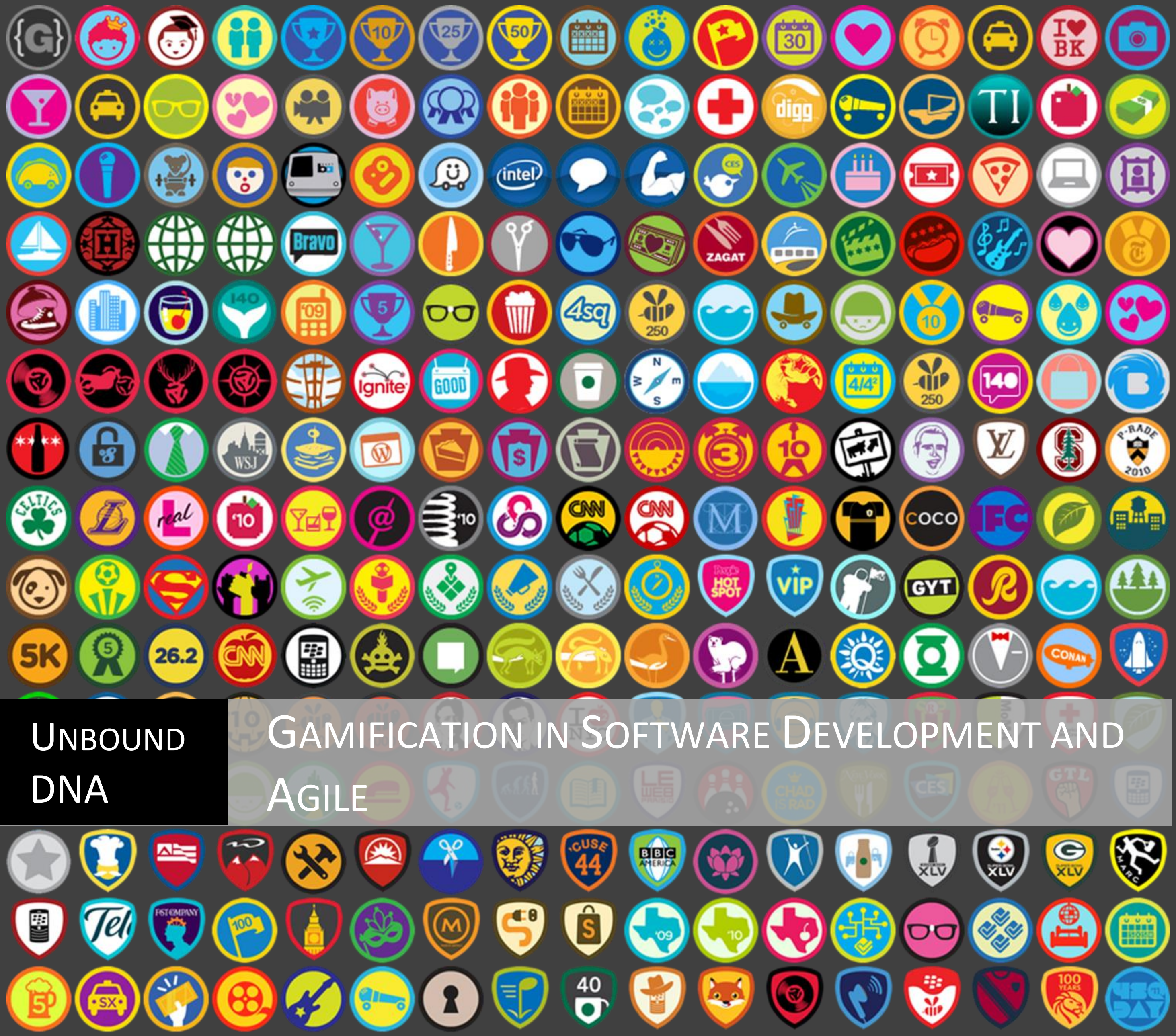


UNBOUND DNA

GAMIFICATION IN SOFTWARE DEVELOPMENT AND AGILE

08 Feb 2014 | Renee Troughton

“Being immersed in a video game, and having your brain stimulated, can encourage creative solutions and adaptions. These beneficial ideas and thoughts can be applied to real life situations. The results can be surprisingly positive for individuals, communities, and the society as a whole.”

- Allen S. Weiss, M.D., president and CEO, NCH Healthcare System

Introduction

When people think of the term 'gamer' it engenders ideas of pasty white teenagers living their days in a zombie like state in the basement of their parent’s house spending over forty hours a week with their eyes glued to the pixels on their screen smashing keys on their keyboard. Despite this stereotype, 72% of American households have at least one digital gamer in their family[1] (up by 5% from 2010) and with the introduction of small games on the iPhone and iPad and through social mediums such as Facebook, the digital gaming door has been re-opened to a whole new audience. Growth of the gaming industry is only expected to rise.

Most of us have actually been gaming ever since we played our first board game, at an age when we didn't know how to read or write. In our early teens we turned to a different form of game, the physical version, sports. In our adult life we have been playing an entirely different and often less motivating game, the political and strategic game, work. If you add up all the time playing the 'work game', the 'sports game' and the digital games on our PCs, smart devices and consoles we are all spending a good portion of our life playing games, some engaging and fun, others less so.

Is work a game?

There are four clear conditions that determine whether an activity is a game or not:

1. There are clear and defined goals or outcomes that needed to be achieved.

2. There are set rules in place to limit how you can go about achieving the goal. By limiting obvious approaches to receiving the goal you are forced to think creatively or strategically.

3. They must provide feedback telling us how we were progressing or when the game is over. Even if we lost they provided us with feedback so that we knew how close we got so that we are further motivated to get there.

4. They must be voluntary. No one forced us to play them.

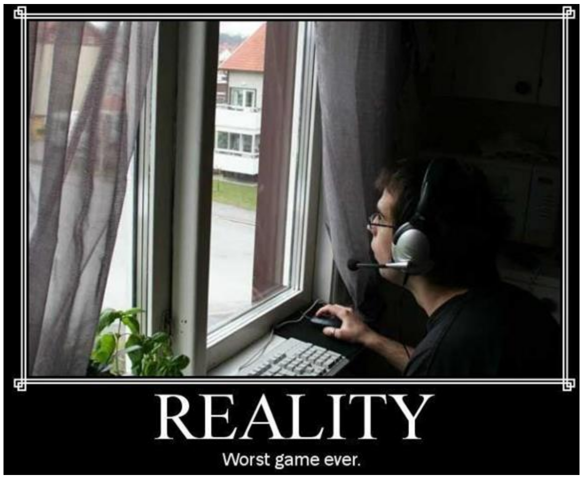
How board games, digital games and even sports fit against these four conditions is quite easy to see. But to see how work fits against these criteria can be a push. No one forces us to work and yet many of us aren't motivated to the same degree of digital games. It is because our hour to hour, day to day activities at work poorly fit against these gaming conditions? We go to work voluntarily but commonly the reasoning is 'to get paid' or 'to pay for the mortgage'. Is it a fiscal or consumerism motivation that forces up to get out of bed and begrudgingly go off to work? Everyone knows that you should do a job you love but with 70% of American works under engaged or actively

[1] Entertainment Software Association – Game Player Data
http://www.theesa.com/facts/pdfs/ESA_EF_2011.pdf

dis-engaged[2] it is suggestive that many of us are only there for the pay packet and not because we love it. This is not a simple problem, this is an epidemic.

At work we will usually have goals but they are set at the start of the financial year and they span twelve months. Consider a game where you get given a set of 'quests' or objectives once a year, I doubt it would be the type of game you would be interested in.

Feedback on our successes or otherwise is similarly as poorly done at work as it is for the setting of goals. Performance reviews are at worst never done, on average done half yearly and at best quarterly. Imagine playing a digital game whereby you get recognition of achievement once a year – no one would play it.



The rules at our workplace are ill defined and often built into the culture and the political environment. Jane McGonigal writes in an book on gaming and its relationship to business and life that “Reality is broken”. She believes it so strongly that she named the book after the belief stating that people are flocking to games because the reality that we live in day to day is not sustaining and motivating us to the same extent as digital games. “Compared to games,” she says, “reality is too easy. Games challenge us with voluntary obstacles and help us put our personal strength to better use.”

It isn't that reality is too easy. It is that the rules are not fixed or that the goals are unachievable against a shifting landscape. Imagine playing a game where when you pressed a button you weren't quite sure whether the cause and effect were stable. You would press the letter '1' expecting to swing a sword in a fantasy role playing game, for example, and instead would jump in the air. The next time you hit '1' you would wrap a bandage on your arm. We need a world that is stable and consistent, where the rules are known, clear and unchanging, a world where we can win.

How does this apply to software development?

This depends on the framework and approach that you take to craft and deliver your software. As a framework, Agile allows for a greatest support against these gaming conditions. Agile at its utter core is a reflective improvement framework. At frequent, regular and defined intervals you inspect and adapt two things – your software and your software delivery process. All of the Agile methods provide a mechanism to reflect, inspect and adapt – in Scrum it is done daily through Burn Down Charts and across iterations through Burn Up Charts, in Kanban it is through Cumulative Flow Diagrams and in eXtreme Programming software craftsmanship is gauged through tools such as code coverage, test case coverage and build statistics. Additionally these methods and ones such as Feature Driven Development inspect the process frequently using techniques

[2] Modern Survey U.S Workforce October 2011 Study http://www.modernsurvey.com/blog/tough-times-call-for-tough-love

such as retrospectives or deltas. Actual output of work is discussed and inspected through Daily Stand-ups, automated testing and end user acceptance testing as each story is rapidly released.

When using an Agile method such as Scrum, work will be broken into commonly two week iterations or sprints. At the start of these iterations there is a clear expectation set of which elements of the software need to be delivered through the stories that are pulled into the iteration. We know that this expectation should be achievable because the amount that was pulled in was derived based upon our previous delivery rate, or in Agile terms, the planned velocity is set to the average velocity achieved of the last three iterations. In this respect Scrum highly supports the gaming condition of clear and defined goals or outcomes that need to be achieved, but for other Agile methods it is more of a stretch. In Kanban, for example, the goal moves away from story completion and more towards reducing the time that it takes on average to get a story through the end to end lifecycle of define, design, build, test and deploy.

The rules that are in place for software development are very interesting. Within Scrum some of these rules are in place, in fact Jeff Sutherland and Ken Schwaber define a simile between chess rules and Scrum[3]. The strong rule that an iteration is a fixed time container that should not be disturbed is a very important and key rule, one that should be considered a limitation imposed, and for good reason as its intent is to provide a safe, unchanging environment even if it is for only a few weeks. Kanban also has rules that are imposed beyond normal software delivery such as limiting work in progress and allowing change through only one (expedite) at a time.

You could say that a Waterfall development method or process also has goals, rules and feedback but its cycles are too long to be considered engaging – the average gamer would be turned off by its lack of real time responsiveness.

Whilst all Agile methods will likely have rules, goals and feedback approaches it is the rapidity and formal regularity of these practices that will always create more engaged developers because of the fact that Agile as a method aligns more often with gaming conditions.

Regardless of the Agile method or even software development approach that you choose to use to develop with - all of them are falling pretty behind on a few aspects of the game conditions. Firstly none of them are truly voluntary. Yes you voluntarily do your job but the story or the task that you pick up isn't chosen by you, nor is the project that you are working on – most developers are assigned to the project. Secondly the rules of political culture within the organization are poorly considered. Despite working inside of a container or having some rules in place to handle change – person or process introduced interference will always impede to some extent. Things won't always go to plan and then more rules are put in place to handle this. Organizational 'value' statements, social contracts and Lean's relentless focus on removing roadblocks are good examples of techniques and methods being applied to resolve this gap.

[3] Scrum Update 2011 http://www.scrum.org/storage/Scrum%20Update%202011.pdf

Can more be done to apply game conditions to the art of software development?

Yes, a lot more. From the way in which we train people, to the way in which we re-enforce desired behaviors or practice compliance we can use gamification techniques more. Gamification is the techniques and mechanics used to solve and engage people through game-like conditions or environment.

The easiest, simplest and most fun gamification technique that you can add is 'Achievements'. When most people think about achievements they think about significant milestones in their life, but in the digital gaming world Achievements are a lot small and often quirkier. They represent the occurrence of positive or negative behavior being demonstrated and re-enforced. Start achievements in one team and watch it spread by the powerful social network tool of word of mouth in the organization. Achievements aren't just for individuals they are also for the team and by introducing them at the team layer you are opening up for cross team rivalry, and positive behavioral control through peer pressure and bragging rights. Other than whether the achievement is associated with the team or the individual there are four key elements to an achievement:

1. The achievement name. The pop culture related or slightly cryptic the better.
2. The achievement trigger. The condition by which the receipt of the achievement is permitted.
3. Positive/negative re-enforcing? You want to encourage the gaining of positive re-enforced achievements and shame the earning of negative achievements.
4. The practice, process, or technique that you are trying to adjust behavior of. You can't just randomly try to adjust behavior; you have to know what behavior you are trying to improve.

Let's take a look of an individual achievement as it relates to Agile:

Showing off!
The person has run the showcase/demonstration three times

Now this is an interesting achievement as sometimes you want to share the demonstration of working software amongst the team, but ultimately you don't want the same person doing it indefinitely which is where this achievement is useful.

So what would a negative focused achievement for an individual look like?

The feet dragger
The person, as a core team member, has skipped five stand-ups in a row

In this example the person who is tracking achievements might need to be pragmatic. It could be that the achiever has been away on sick leave for three weeks, but the intent is clear – this person has either dropped the ball or they are focusing their attention in the right areas.

From a team perspective there is a lot of fun achievements that you can create, here are just a couple:

Venus
A burn up chart of the team's velocity against expectations has been

produced and made visible (you’ve got it!)

Back to the Future
Retrospectives have been held at the end of each Sprint for five Sprints in a row

I see dead walls
Less than 5 cards move inside of a Sprint (or within 2 weeks if using Kanban)

Within an Agile team the Scrum Master or Iteration Manager would be the person that would keep a track of achievement progress against an individual and team leaderboard. Paper awards or appropriate symbols related to the achievement earned can also be given.

Gamifying the software you are developing

Along with gamifying the way in which we are working as team we can also keep in the forefront of every software developers, business analyst and product owner's mind the question “How can I build a product that not only meets the needs the business' outcomes but also enables the customer to feel like they are playing a game, that there is fun and a sense of achievement built in?”

The marketplace is growing quickly with internet applications that are taking in elements of gaming such as social networking, team-based cooperative play, strategy, role playing, achievements or competitive score boarding. One great example includes Red Critter Tracker which allows you to manage your business's projects within an in built organizational and project social network engine, achievements and reward. Another great example is Rypple, a people performance management tool, which enables a similar set of gaming features as Red Critter Tracker but focuses more on the person than the project.

It isn't just the small-medium or start-up companies that are looking into this. Big organizations such as SAP are trying to enter the foray by literally trying to build games into the software.

Conclusion

Whether you are building software or managing a software development team, gamification can be a useful set of techniques to make work more engaging and encourage the change of behaviour for both those that you work with or the customers that use your systems.

About the author

Renee Troughton is an Agile Coach with over 14 years of experience in software development, coaching and organizational change. As both a Scrum Master, Agile Coach and Agile Trainer, Renee has spent many years working with teams to helping to transform behavioral change through gamification.

Renee Troughton is Director and co-founder of Unbound DNA and can be contacted at renee@unbounddna.com or through twitter at @AgileRenee.

Unbound DNA

Deliver. Now. With Agility.

http://www.unbounddna.com