

## Managing Projects in a Games Factory: Temporality and Practices

Patrick Stacey

*Innovative Manufacturing Research Centre,  
School of Management, University of Bath,  
Bath, BA2 7AY, UK.  
email: mnppks@bath.ac.uk*

Joe Nandhakumar

*School of Management, University of Bath,  
Bath, BA2 7AY, UK.  
email: j.nandhakumar@bath.ac.uk*

### Abstract

*The online gaming world is growing voraciously and so too is the wider market for computer games which now exceeds the annual global revenues of cinema [46]. Even so, we have only a limited scholarly understanding of how games studios produce games. Games projects require particular attention because their context is unique. We aim to elucidate the games development process from a time perspective and suggest ways of managing it. We argue that a games project is characterised by a compound of temporal rhythms that emanate from the individual, team, company, industry, nation, inter-nation and marketplace levels. Project managers need to take account of these 'inner' and 'outer' temporal rhythms since they can affect development culture and project chronology.*

### 1. Introduction

The UK government once hailed Lara Croft (Tomb Raider) as an icon of "Cool Britannia" and as exemplifying the role of innovative design in a knowledge-driven economy [47]. The iconoclasts must have been 'waiting in the wings'; Lara's British creators recently lost their development contract to an American studio after defaulting on numerous project milestones and ship-dates. Such events have propagated wide concern at practitioner conferences [48] and in government agency reports [1] over how to manage computer games projects. While much academic research focuses on issues like user/player behaviour (obsession [3], aggression [4], acceptance [5], virtual social conflict [6], and real-life mimicry [7]), game design (techniques such as scenarios, body storming, paper prototyping, and rapid prototyping [8-12], simulation [13]) and artificial intelligence (the search for more intelligent 'computer' behaviour to enrich the single-player gaming experience [14]), we have only a limited scholarly understanding of the work and management practices of games

development organizations, although we note the paper by Baba and Tschang [15] which examined the development of games at Sony from the perspective of innovation.

The state of the art is for a games publisher to define a chronology of milestones for the games studio. This chronologicistic approach to managing projects is quite standard across many organizations, although it does not appear to have been reliable. Why not? We argue that games projects, particularly those perceived to be 'innovative' or ground-breaking (Sony's Gran Turismo for example), are largely characterised by unscheduled improvised activities. Therefore, traditional schedule-driven project management approaches may not be appropriate in games projects. The more fundamental reason for this is that the context within which the games development process takes place is very different to that of other software projects.

Computer games (including those played on PCs, consoles, and hand-held devices like mobile phones) are designed to provide an entertainment experience, unlike a traditional software application which is designed to provide a 'solution'. Computer games are played in order to satisfy an *emotional* and sometimes intellectual need, unlike a traditional software application which is used to satisfy a *functional* need. Courses on games design even teach 'emotioneering' [49]. Playing computer games is compelling, sensory [15][36] and whimsical (in the moody and extemporaneous sense) occurring within the context of 'play' as opposed to 'work'. Indeed, throughout a typical day, a person may drift between modes of work and play – particularly given today's availability of mobile games. In a multiplayer mobile game, people take on *alter egos* and connect with a community of players out of their own free will as opposed to using mobile business applications (for example) to contact people out of obligation.

The whimsical nature of computer games entertainment represents a social context which distinguishes, enables and constrains the computer games project, principally in two ways. Firstly, in the moody sense of the word, since games are whimsical, they are not produced in accordance with the specific needs of a

specific player – there are no user requirements as such [15]. Thus the relationship between developer and player-user is weak. Unlike in more traditional and even agile project environments into which customers may be ‘implanted’ [37], and whose needs are made explicit. A games project’s so-called requirements instead reside with a visionary designer who interacts with the ‘fuzzy’ trends and tastes of popular culture; “what’s cool”. Secondly, using the temporal connotation of the word, since games are whimsical, designers do not necessarily follow a strict development chronology. Designers may prefer to develop a game in a situated, ‘bursting’ manner as they are influenced by and influence popular culture.

Nonetheless, managers of software projects in general share a common ultimate temporal aim to some degree; to ‘deliver’ at a point in time that aligns with an event (or events) outside the development context - the Christmas season for example. Project delivery is therefore an important issue even in so-called whimsical games development. However, we must first surmount our limited ex tant scholarly understanding of the games development process before we can address such an issue. Therefore, our aim in this paper is two-fold: (a) to elucidate the games development process from a time perspective by drawing on an in-depth study in a games development organization, and (b) to outline guidelines for managing games projects taking into consideration the many temporalities that exist inside and outside such projects.

## 2. Theoretical foundations

A popular perspective on time is one which is predominantly objective. Characteristic of this view are metaphors such as the line, cycle and clock [22]. With regards to the linearity-objective view, De Grazia [38] sees the straight line metaphor of time as being born out of Christian society in terms of the pathway from sin to salvation. This is perhaps the most persistent metaphor in domain unspecific approaches to project management such as PERT, GANTT, and Critical Path Method. Intrinsic to ‘line’ is sequence and linearity; notions evident in threshold process models [39] such as the Capability Maturity Model and stagewise structured development methodologies [2], although, to varying degrees, the cycle and line co-exist in stagewise methodologies. Time is thus seen as unitized, homogenous (an hour in the morning is equivalent to an hour in the evening), measurable [40] and therefore predictable [18]. Thus, the line metaphor ultimately structures the planning experience giving the impression that work and time may be decomposed and then re-composed in order to derive project milestones. Time is seen as mind-independent. Software projects then are

typically characterised by an objective perspective of time.

The anti-thesis is that time is subjective, i.e. it is personally constructed [22], heterogeneous [40], non-linear, and non-sequential. Individuals exercise autonomy to self-determine their time patterns; every day examples may include freelancers and the self-employed. Autonomy connotes self-empowerment and control. An autonomous mode of working has been observed in certain internet software development situations [41] where developers were empowered as decision-makers of product features and schedules. However, this was more concerned with the autonomy of a team as opposed to an individual, being the focus in the subjective view of time. Autonomy does not necessarily connote capability; one may possess the opportunity to self-determine one’s time patterns but exercising it is not a corollary. Therefore, we can separate the ‘opportunity’ to self-determine time [42] from its realization, i.e. separate the intention to do something from the act [43]. Human action or agency is an important idea in the discussion of subjectively experienced time, as it is in the intersubjective view of time.

A third view then, is that time is experienced intersubjectively through a process of temporal structuring [20][21][22]. This view is thought to constitute a bridge between the objective and subjective ways of thinking about time [22]. By drawing on social theories we consider time as constitutive of forms of social activities [20]. Time in the day-to-day experience is therefore constituted in repetition; activities recurring in temporal locations in the form of routines, phases and project deadlines (for example). These recurrent practices help to produce and reproduce “temporal structures” [21, 22]. Temporal structures or rhythms are therefore individuals’ everyday practices that have been institutionalized through the reproduction of such practices. Under different conditions individuals reinforce or change their temporal structures, as well as introduce new ones. Individuals have the potential for reinforcing and altering temporal structures although they are not always aware of it [20]. Thus, temporal structures serve to constrain or shape the actors’ activities (structure as medium) and at the same time enable or be shaped by human social action (structure as outcome). Thus a project deadline may be viewed as both enabling and constraining of human action. Our analysis draws on temporal structures as an initial sensitizing device [23] to structure the interpretation of our data.

We hope this perspective will make a fresh contribution to the debate on managing software projects, not to mention managing the games development process.

### 3. Research setting and method

We conducted an in-depth study of a computer games studio in Singapore, a country which has little in the way of natural resources and therefore relies on external trade and foreign investment to buttress its economy. In order to attract foreign investment, government agencies such as the Infocomm Development Authority (IDA) produce statistical reports and success stories which are presented at international games events such as E3. Local companies (games studios and otherwise) are encouraged and at the same time under some pressure to perform and deliver so as to bolster the country's portfolio of achievements and hence its attractiveness. Time is of the essence in this agile economy, especially in the wake of economic and social crises such as the Asian Financial Crisis and SARS. Although the computer games industry is a mere fledgling it has been identified as one of the new 'engines' of economic growth by the government who speak of developing games 'factories', 'highways' and 'digiports'. In these industrial metaphors, there appears to be an implied desire to ramp-up games development to the point where games are rolling off a conveyor belt somewhere; a Fordist notion of time at a national level.

The research approach adopted in this study is interpretive [23] involving a collection of detailed, qualitative data on the games development process in its context. We conducted twenty interviews with various local games studios in Singapore to try to understand how they manage their games development process. Put another way, how do games studios in Singapore manage their software development process? In this paper we focus on one studio, Computer Games Studio (CGS – a pseudonym) as an example.

CGS is a two year old company with no more than twenty staff, although this headcount fluctuates somewhat due to staff turnover. Even though the company has its main operations in Singapore, it has developed numerous games in association with studios in France and Italy, and distributed them throughout Europe. It is further recognised by Singapore's Infocomm Development Authority (IDA) as one of the leading local studios.

#### 3.1. Data collection and analysis

From January 2004 to April 2004 we made several visits to CGS and conducted interviews with various people in the company. Detailed notes were kept on the interviews and the observations of work practices during each visit. Table 1 exhibits the people interviewed and their position at CGS.

**Table 1. CGS site visits**

Nature of Interview	Position of Interviewee
Group interview	Group
Unstructured interview	Project Manager
Unstructured interview	MD
Unstructured interview	CG programmer
Unstructured interview	Lead programmer
Structured Interview	Intern
Structured Interview	MD
Structured Interview	CG programmer
Structured Interview	Project Manager
Structured Interview	Lead programmer

Further, company documents and flowcharts (as delineated in section 4.1.1) were also analyzed to gain more insight into the practices observed and to verify interview notes. By adopting an interpretive approach to collecting and analysing the interview data, we attempted to understand the phenomenon (the games development process) through the meanings that team members assign to that process. These meanings may be rooted in data incidents like jargon, symbols, and metaphors the interviewee uses. It is up to the researcher to explore these 'incidents', taken-for-granted understandings [24] within the setting.

To aid the analysis, the interview transcripts were imported into nVivo – a Computer-aided Qualitative Data Analysis Software (CAQDAS). nVivo is simply a tool for organizing, structuring and thinking about the data; the analysis is ultimately done by the researchers not the tool. The analysis was guided by approaches described by Strauss and Corbin [25]. Their approach should not be construed as a structured methodology but as a portfolio of principles and techniques the researcher internalises and brings to bear on the data.

To fully *digest* the data, a microanalysis of words, phrases, and lines was carried out. The micro-interpretations this produced were stored as 'databites' in each rich text document so they could be conveniently revisited later to inform the bottom-up conceptual development. This data mining was free flowing and stimulated a number of interpretations; an important principle in research such as this [23]. This process produced text per document which was of almost equal length to each narrative, i.e. it was extensive. Next came the low-level coding of the text. The 'tags' were based on actual words, concepts in the text, i.e. coded *in vivo*.

Staying close to the text was an attempt by the researchers to prevent them from being seduced into making premature leaps towards higher-order concepts that are predicated on the researchers' own mental models. The research process is perceived as a means of vocalising and elucidating the interviewees' mental models [26]. Altogether, 306 'free nodes' or 'open codes' were generated. These needed to be structured to facilitate

the discovery of patterns in the data. To this end we employed structuration theory [20] as a sensitizing device; the free nodes were categorized as actions, intentions, intended outcomes, and unintended outcomes. Links were then established between nodes across these categories to form a ‘chain of evidence’, e.g. actions and their (un)intended outcomes.

#### 4. Case description

CGS has a management hierarchy with four layers of managers: senior management, middle-management, project leaders and executives. However, since the co-founders are spread across the top three levels these tend to equivalence. The working environment at the premises gave a sense of home-from-home: there was a meeting room with antique Indonesian furniture which doubled as the Managing Director’s office, an office for the more technical members of the development team who seemed subdued although their walls seemed to reflect their fantasies, an office for the artists who had various traditional games like table football set up, and a laboratory full of computers for rendering purposes. The environment seemed conducive to long hours of work.

The study mainly focused on the work practices of the managing director, a project manager, the lead programmer, a computer graphics (CG) programmer, and an intern programmer (Table 2). These appear to be the principal organizational actors involved in games projects. Our future research will endeavour to widen the base of interviewees.

The everyday practices or temporal rhythms of the managing director, project manager and lead programmer are stretched across several roles. The managing director, Alf participates in business ‘events’ outside the organizational context with government agencies, shareholders, partners, clients, and distributors.

**Table 2. Organizational actors**

<b>Alf</b>	<b>John</b>	<b>Richard</b>
Managing Director <i>Co-founder</i>	Project Manager <i>Recently joined from IBM</i>	Lead Programmer <i>Co-founder</i>
<b>Mac</b>	<b>Angelina</b>	
CG Programmer <i>Co-founder</i>	Intern programmer <i>Recently joined from France</i>	

Inside the context, as well as overseeing the company, he performs the function of game designer. This role typically involves the conceptualization of a games project. John described himself as “wearing several hats”; a role description that resonated with Alf’s. His official designation is ‘project manager’ although he told us he is also a programmer, a process ‘cop’, and quality engineer.

Richard’s three primary responsibilities as lead programmer are to oversee the programming team, integrate the artwork, and advise on the feasibility of the game designer’s ideas; he referred to this as “sizing down” ideas, which makes him feel rather like a ‘whistle blower’. The CG programmer, Mac, focuses on adapting graphics models (sprites) for the medium, and adds the textual program “codes” that define their behaviour. The intern from France, Angelina, concentrates on the “raw coding” of games, although she is asked to write design documents occasionally.

#### 4.1. Perspectives on CGS’ development process

Our first visit to CGS comprised a meeting with the organizational actors listed in table 2. During that meeting we asked the group to describe their development process for their most recent games project.

**4.1.1. Consensual perspective.** Led by the managing director, we documented the process in our field notes as: “Alf described how the process began with the conceptualization of the game. He trawled the internet for the look and feel before putting down his ideas into a design document (DD); we were shown an example of this. The DD was then circulated throughout the company for comments. The document was circulated and modified until a ‘stable’ concept was reached, i.e. all developers were happy with it. Then Richard translated the DD into software modules and assigned them to the various engineers. Once each engineer had done their respective coding and unit testing they integrated their modules and the game as a whole was assembled. Integration testing and fixing followed. Once they had eliminated the major integration bugs, the system or game as a whole was tested, i.e. black-box testing”.

During subsequent visits, we asked the interviewees to prepare flow diagrams of their development practices for their most recent games project; an approach based on Radice et al’s ([27]: 100) in their programming process study. The interviewees produced illustrations of varying detail and content which we have summarized in tabular form (Table 3). The intern produced the richest flow diagram (Figure 1). Her perspective was an essentially chronological one, i.e. it depicted a chronology of stages or processes. The only people mentioned were artists. The project manager’s and managing director’s perspectives were also chronological, although the MD did mention the producer’s liaison with the programming, art and sound teams. Interestingly, the lead programmer and CG programmer diagrams did not depict a chronology of linear stages or processes but of inter-actions between people.

These perspectives formed the basis for further discussions with each participant during which we sought

to ‘unpack’ and explore the meanings of their terms such as ‘testing’. This terminological ‘unpacking’ also helped us understand the intentions, actions, and outcomes that lay behind them. On a practical level the exercise helped the interviewee to focus and bring to mind their working context.

**4.2. Documentation and tools**

According to the interviewees, two documents were produced before the game went into production, which were referred to as the design document (DD) and technical document (TD). The design document took the games designer months to produce, relying on a confluence between inspiration, market trends, and demands on the games designer’s time. The technical document or specifications were based on the content of

the design document and took the lead programmer a few weeks to compose. The implementability of game features were discussed and negotiated with the games designer. Once the specification was agreed, the technical team set to work with the aid of development tools installed on office PCs such as Jcreator (an Integrated Development Environment for Java programming), Wireless ToolKit, a mobile phone screen emulator (provided by the respective manufacturer to test Java games), Java SDK (the standard compiler for Java), J2ME (Java 2 Micro Edition – a ‘lite’ version of Java SDK). The Art team used 3Dstudio MAX for animation and Adobe Photoshop (a bitmap graphics editor) to create concept art, in-game art (sprites) and animation. The nature and physical location of these tools meant that the social and temporal organization of work was decided *day by day* within the offices of CGS.

**Table 3. Actors’ perspectives on CGS’ process**

<b>Actor</b>	<b>Flows</b>
Intern (see Figure 1)	Design phase → design document → decompose game into modules → unit testing → feedback between software engineering, programming and project management → changes to architecture and schedule → modules implemented → modules integrated → modules debugged → game stabilized → game ported (e.g. to console, mobile device).
Project Manager	Concept → Design → Implement → Testing → Market/Distribution.
Managing Director	Conceptualization → Documentation → Project management → Marketing → Producer (liaise with programming, art and sound team) → QA → Distributors.
Lead Programmer	Game Designer ↔ Programmer ↔ Artist.
CG Programmer	Artist/Animator ↔ Programmer.

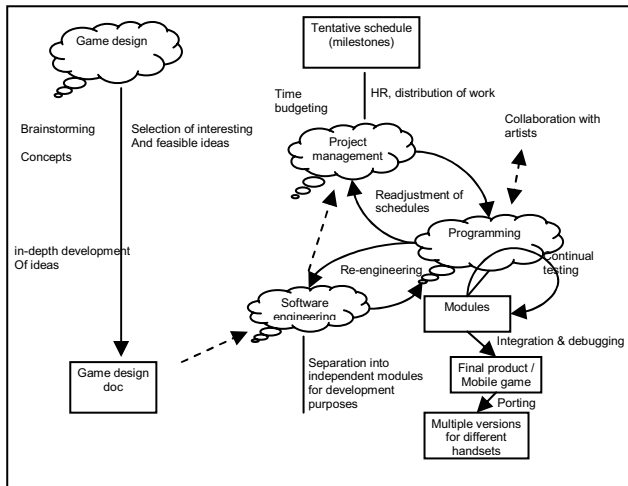


Figure 1. Intern's flow diagram

#### 4.3. Play-testing subcycle

Throughout the project's lifetime, the game was continuously play-tested in order to (a) evaluate its aesthetic appeal, and, (b) to "witch hunt" (John's expression) bugs. Determining the aesthetic appeal of the game involved a subcycle akin to a hermeneutic circle – playing the parts (or modules) and then judging how they interact as a whole. Play-testing was conducted until the point when a (subjective) product evaluation could be made, i.e. an opinion was formed. Invariably, this led to changes in the way the game was conceived to some degree. As a result, milestones ended up being 'massaged' and/or long hours had to be devoted by the developers in order to compensate for the scope-creep. Therefore, we conceptualise the play-testing subcycle as a rapid feedback between (re)conceive, (re)design, (re)code, (re)play-test, and (re)evaluate (Figure 2). The overall project cycle and play-test subcycle constituted temporal structures.

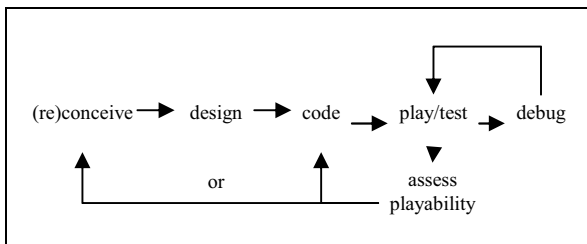


Figure 2. Play-test Subcycle

However, the subcycles' duration, frequency, and reconfiguration were decided daily by the team members depending on the circumstances facing them; therefore these temporal structures were being produced through improvisations as opposed to being reproduced through

routine actions. For instance, as 'bugs' arose during play-testing they were not seen by all as 'errors'. The CG programmer subscribed to this view and explained that bugs inspired new creative dimensions. He recounted one example to us: during play-testing the lead programmer found a place in the game-world where his character could 'camp' without being shot. Instead of seeing this as a 'bug' however, it inspired the team to create an additional enemy that uniquely had the range to reach that part of the 'world'. Whether such a 'feature' would be added to the game was ultimately the game designer's decision. However, due to business commitments (he was also the MD) with government agencies, and shareholders he was largely unavailable to make such decisions. This dependency led to waiting and stalled development. However, the developers began making their own design change decisions; a form of self-empowerment. A similar dependency was made known to us with artists who were perceived by the engineers as 'bottlenecks' in the process; "artists could not work to the deadlines", we were told, and when they tried to work to them the standard of their art work fell. The CG programmer, who worked closely with the artists thought the engineers should learn graphics tools such as Adobe Photoshop so they could at least produce their own test graphics; tools were seen as enabling skill. It was thought that this would reduce the engineers' dependency on the artists. However, the lead programmer had already begun finding his way round Photoshop on top of his other duties; according to the MD Richard was a games designer, project manager, quality assurance engineer, programmer and overseer of interns (different from Richard's account). Artists were perceived not just as 'bottlenecks' but as generally being highly 'mobile'. According to Richard a string of artists had joined and resigned from CGS, perhaps because they did not have "the right knowledge-level or understanding [of games development]" he mused. By this he meant a grasp of terms like MIDP (Mobile Information Device Profile). Consequently many freelance artists had to be recruited to fill the 'voids' left by resigning artists. This compensatory strategy was problematised by the MD 'brokering' between the freelancers and the core development team, leading to mis-communication and either the over or under production of artwork. This mobility of staff was made known to the researchers with respect to interns too. Interns were usually attached for only three months. There was some reliance on them to do the "tedious" programming tasks which were executed with increasing productivity towards the end of their attachment. Learning curves were revisited and productivity fell each time one intern arrived and another left. Recruiting interns from overseas was a further bone of contention. They had one from Germany but his inability to articulate his problems to the rest of the team was blamed for the project's delay.

## 5. Analysis

Temporal re-structuring during play-testing was characterised by rapid feedback between the activities of (re)conceive, (re)design, (re)code, (re)play-test, and (re)evaluate (Figure 2), and was subject to daily reconfiguration by the team members depending on the circumstances facing them. These development activities were contiguously temporally located with no real sense of chronology. Such games development activities may be said to be situated, with no definite plan, occurring ‘in the moment’. Within the context of the project, the emerging circumstances were self-produced; by the actors’ evaluation of the game’s aesthetics, i.e. game-play, graphics, sound, and narrative. We may think of aesthetics as surrogates for the intended and unintended outcomes of the developers’ actions; the intended outcomes pertaining to those (standard) features which were planned (e.g. game lobby, high score, game objective), the unintended ones being unplanned and emergent from the development process as described above in the case. Thus, the intended and unintended outcomes of the developers’ actions were embodied in the game, as a temporal structure, and served to constrain, inspire and enable the ongoing activities of the developers.

As Ciborra argues, there is another connotation to situated action besides its temporal one. Situated action also connotes emotion [28]. This is consistent with what we found in the context; the passion and moodiness of the artists and programmers when their temporal rhythm was interrupted by the researchers and in their vocal attitude towards ‘process’, “its about the product, the game-play, not the process”. Thus, the emotion of the developer also affects their temporal structure, in terms of feeling like doing something and doing it on the spur of the moment, extemporaneously; an impromptu or improvised action.

Another form of improvised action is the make-do variety [28] which we also perceived as occurring in the context. As actors left the context permanently (by resigning) and/or temporarily (e.g. absence of the games designer), incumbents had to fill the gaps or roles they left behind, whether it was one of their ‘competencies’ or not, which they did through improvised make-do actions. Therefore, multiple temporal rhythms developed within the actor since each role (artist for example) comprised categories of time-demanding activities (concept art creation, wire framing) and in effect constituted a subculture.

Actors joining the context also influenced the incumbents’ temporal rhythms. For example, John had only just joined CGS from IBM where he used to develop banking systems. He described the development culture at IBM as: (a) “the specs were very thick to prevent scope

creep”, (b) the development process was very structured, (c) most of his time was occupied with writing documents, (d) it could take two weeks just to change a single line of code. He had been used to a development environment where his work activities occurred in a pre-determined chronological order. Through the introduction of a suite of documents (concept, design and technical) he intended to ‘free up’ time, as well as enhance productivity and improve guidance for the interns. To mobilise his intentions he constructed symbols that would appeal to management in the form of proposals. We interpret this as (unwittingly perhaps) trying to reconstruct the temporal rhythm he had been used to at IBM in CGS, i.e. to reproduce a previously experienced temporal structure.

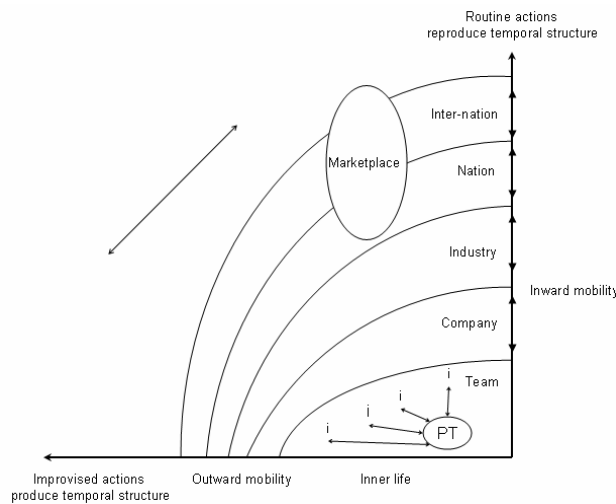
Outside the CGS context, the mobile phone manufacturers, the Singapore games industry, the international games industry, and the marketplace all have their own temporal rhythms or schedules which impact the temporal structure inside the context to some degree. At the industry level, most manufacturers take weeks or months to provide the unlock-tools that CGS needs in order to conduct system testing of mobile games. One particular manufacturer provides the tool almost instantly to CGS. Consequently most mobile games are produced for that manufacturer’s phone; the temporal rhythms of CGS and that manufacturer may be described as being in-sync. At the government level in Singapore local studios are encouraged to align with a national cause and calendar, i.e. to make Singapore an attractive business destination through participation in ‘recognised’ national (Games eXchange Alliance, IGDA Singapore Chapter) and international games events (Games Developers Conference, E3). Fortunately, there is some temporal synchronicity as far as international games events are concerned; most studios and games bodies around the world try to showcase their games or capabilities at E3 which is to games what Cannes is to film. The consequence of attending such events is that developers are spatially removed from their development contexts. Of course people can work online these days but there are some activities which are more effectively conducted in context; managing, leading, and sharing for example. One consequence is that this temporary mobility and absence feeds the *make-doism* of the team.

## 6. Discussion and implications

We have elucidated the complex and messy nature of the temporal rhythms that affect a games project. A games project then, is characterised by a compound of temporal rhythms or structures that emanate from the individual, team, company, industry, nation, inter-nation and marketplace levels.

Figure 3 depicts how these multiple temporalities shape each other. The vertical axis in this figure

represents the incoming or inward mobility of staff which we associate more with the reproduction of structure. The horizontal axis represents the outgoing or outward mobility of staff which we associate more with the production of structure. *Individuals* in a team produce temporalities through improvisations which arise through play-testing (PT), the exiting of staff from the context (outward mobility) and the developer's inner-life or mood. *Companies* attempt to shape the temporalities of the *team* by forming intentions (i.e. planning) and acting upon them through the construction of symbols with such legitimating intended outcomes as 'freeing up' time. Within the *national* context of our study companies and *industries* appear to conform to or act in accordance with government policies, i.e. their temporalities are shaped by those of government. Within the national context of our study, the *nation* appears to be shaped by the temporality of that economy and *marketplace*.



**Figure 3. Multiple temporalities**

Although an *international* temporal rhythm may seem far removed from that of the development team it is clear from our analysis that they are not. The pressure for a games development team to reproduce temporal structure therefore starts at a high-level and permeates down through government policies, industry calendars, labour market (particularly as managers join companies – inward mobility) and the management strata of a company. Therefore the reproduction of structure is linked to power. We observed an example of this inside CGS when the newly appointed project manager (John) began forming intentions and acting upon them through the construction of managerial symbols with such legitimating intended outcomes as increased productivity. He therefore, perhaps unintentionally, attempted to reproduce a temporal structure experienced outside the context during the course of his professional history. Consequently, while management shared and agreed with the project

manager's intended outcomes for a new 'suite' of documents, the developers did not. As the incumbent developers recognized the implications of such intended outcomes on their sedimented habits, routines, so they too mobilized their own contexts to counter the intentions of the project manager, "it will take the creativity out of the process" (Richard). In the end, whether the project manager's actions changed the structure depended on which subculture could legitimate their claim the most. In turn, this depended on who had the better understanding of their context and how to mobilize it. Therefore, an unintended outcome of the project manager attempting to alter the temporal structure of a context in the form of another was a power struggle characterised by the trading of symbols between subcultures. In summary, we can think of the team as a core within which temporal structures are being produced but encapsulated by the magma of temporal structure reproduction.

Our theoretical insights into the inter-acting multiple temporalities that affect a games project have important implications for practicing project managers. Our findings suggest that a project manager needs to be aware of the various temporal rhythms that affect their development culture at the levels of the individual, team, company, industry, nation, inter-nation and marketplace. For instance, an awareness of individual, team, and industry partners' temporal rhythms can help to anticipate possible bottlenecks and at the same time identify opportunities. However, depending on the nature of the game, i.e. how innovative it is, the temporal rhythms of the team may be frequently reconfigured through the improvised actions of developers making them difficult to anticipate. Improvisation, venturing, and intuitive action are the preferred course of action under these circumstances, i.e. when the game is novel and unclear [29]. We found that play-testing inspired more ideas and was central to the creative process. A consequence however was project-delays although the game features were enriched. The question is how to reach creative closure on a game at a point in time that aligns with an event (or events) outside the development context - the Christmas season for example. One may infer that by encouraging even shorter sub-cycles yielding more play-testing, this could lead to more frequent product evaluation, and consequently heighten the intensity of improvisation. By doing so perhaps the creative team members would reach a point of saturation or burn-out, and consequently the game reach a point of creative maturity. In this situation 'champions' and 'heroes' are paramount to project success. Champions are able to deal with the complexity of the situation as it emerges [30], integrate teams of people [31] and earn the respect of fellow developers [32]. The grooming, retention, and hiring of such people could be viewed as strategic. Under the circumstances of a sequel to a game being developed, the situation is clearer and a less



improvised, more chronological, planned approach to project management may be appropriate [29]. However, even games franchises such as Gran Turismo (1, 2, 3, 4) and Tolkein Games (The Hobbit, The Lord of the Rings: War of the Ring, Middle-Earth Online) have limited shelf-lives which means after periods of emerging clarity the studios plunge back into the foggiest of innovation.

Being aware of the various temporal rhythms that affect their development culture is one thing, but understanding how to synchronize them is another. For those ‘outer’ temporal rhythms (e.g. government, international) that appear out-of-synch with the ‘inner’ ones (e.g. team) a decision will need to be made on the strategic importance of synchronizing them. If deemed strategic, this may entail ensuring the team can operate autonomously and make their own design decisions in the absence of managers participating in these ‘outer’ temporal rhythms (outward mobility). Therefore, the effects of synchronizing inner and outer temporal structures is that both will change depending on the situation.

## 7. Conclusions

In this paper we offered an insight into the games development process from a time perspective by drawing on an in-depth study in a games development organization. We illustrated that a compound of temporal structures emanate from the individual, team, company, industry, nation, inter-nation and marketplace levels. We outlined some guidelines for managing games projects, taking into consideration the many temporalities that exist inside and outside such projects. Our findings further suggest that the participants are very much ‘alive’ in a games development project and shape the temporal structure as much as being shaped by it through their ongoing improvisations. Some would style such contexts as “immature”, and “reactionary” [33]. However, messy, on-the-fly, real-life decisions [29] are a norm, while structured software development methods seem to be methodological fiction [34]. Such approaches view time as mind-independent, homogenous, and as providing an objective framework for the completion of the tasks. The desire to be chronological, to reproduce temporal structure at all the levels we discussed seems to be linked to the construction of symbols of quality [35] and professionalism that appeal to investors in companies and nations. Inside the core development team things look rather different however, depending on the nature of the games project. In innovative games projects, where the situation is less clear, improvisation, ‘champions’ and ‘heroes’ are paramount to project success. Such people are able to deal with the complexity of the situation as it emerges [30], integrate teams of people [31] and earn the respect of fellow developers [32]. Following their seminal

programming process study (upon which the SW-CMM was based) Radice et al ([27]: 99) concluded that “...we learned once again that the most valuable resource on any project is the people who live with that project on a daily basis...” . When and *if* the situation is clearer, the use of structured methods and process models may be more appropriate. However, such approaches should look beyond the boundaries of the project to take account of the various inner and outer temporal rhythms that affect development culture and project chronology.

We anticipate that our findings have applicability to software projects operating within a creative context and bearing similar process traits to those that we found such as improvisation.

## 8. Acknowledgements

The financial support by the UK Engineering and Physical Sciences Research Council and the Innovative Manufacturing Research Centre (School of Management, University of Bath) is gratefully acknowledged.

We are also grateful to the comments of official and unofficial reviewers of this paper.

## 9. References

- [1] Strategy Consultants and Department for Trade and Industry (2002). From exuberant youth to sustainable maturity: Competitiveness analysis of the UK games software sector. London.
- [2] Boehm, B.W., A spiral model of software development and enhancement. *Computer*, 1988. 21(5): p. 61-72.
- [3] Rehak, B., Mapping the Bit Girl., in *Information Communication & Society*. 2003, Routledge, Ltd. p. 477-496.
- [4] Williams, R.B. and C.A. Clippinger, Aggression, competition and computer games: computer and human opponents. *Computers in Human Behavior*, 2002. 18: p. 495-506.
- [5] Hsu, C.-L. and H.-P. Lu, Why do people play on-line games? An extended TAM with social influences and flow experience., in *Information & Management*. 2004. p. 853.
- [6] Lastowka, F.G. and D. Hunter, The Laws of the Virtual Worlds., in *California Law Review*. 2004, California Law Review. p. 3-73.
- [7] Nutt, D. and D. Railton, The Sims: Real Life as Genre., in *Information Communication & Society*. 2003, Routledge, Ltd. p. 577-592.
- [8] Clanton, C. An Interpreted Demonstration of Computer Game Design. in *ACM CHI 98*. 1998.
- [9] Johnson, D. and J. Wiles, Effective affective user interface design in games., in *Ergonomics*. 2003, Taylor & Francis Ltd. p. 1332.
- [10] Cheok, A.D., et al., Touch-Space: Mixed Reality Game Space Based on Ubiquitous, Tangible, and Social Computing. *Personal and Ubiquitous Computing*, 2002. 6: p. 430-442.

- [11] Bjork, S., et al., Designing Ubiquitous Computing Games – A Report from a Workshop Exploring Ubiquitous Computing Entertainment. *Personal and Ubiquitous Computing*, 2002. 6: p. 443–458.
- [12] Rouse III, R., Everything Old is New Again: Remaking Computer Games. *Computer Graphics*, 1999. May 1999.
- [13] Kaneva, K. and T. Sugiyama, Design and simulation of interactive 3D computer games. *Computers & Graphics*, 1998. 22(2-3): p. 281-300.
- [14] Schaeffer, J. and H. Jaap van den Herik, Games, computers, and artificial intelligence. *Artificial Intelligence*, 2002. 134: p. 1-7.
- [15] Baba, Y. and F.T. Tschang, Product development in Japanese TV game software: The case of an innovative game. *International Journal of Innovation Management*, 2001. 5(4): p. 487-515.
- [16] Johnson, J.R., The software factory: Managing software development and maintenance. 1991, London: Butterworth.
- [17] Debou, C., A. Kuntzmann-Combelles, and A. Rowe. A quantitative approach to software process management. in *Software Metrics Symposium, 1994., Proceedings of the Second International*. 1994.
- [18] Paulk, M.C., et al., Capability maturity model, version 1.1. *Software*, IEEE, 1993. 10(4): p. 18-27.
- [19] Boehm, B.W., *Software Cost Estimation with CocomoII (with CD-ROM)*, 2000, Upper Saddle River, N.J.: Prentice Hall.
- [20] Giddens, A., *The Constitution of Society: Outline of the Theory of Structure*. 1984, Berkeley, CA.: University of California Press.
- [21] Nandhakumar, J., *Managing Time in a Software Factory: Temporal and Spatial Organisation of IS Development Activities*. *The Information Society*, 2002. 18(25): p. 251-262.
- [22] Orlikowski, W.J. and J. Yates, It's About Time: Temporal Structuring in Organizations. *Organization Science*, 2002. 13(6): p. 684-700.
- [23] Walsham, G., *Interpreting IS in Organisations*. 1993, Chichester: John Wiley.
- [24] Schutz, A., *Phenomenology of the Social World*. 1967, Evanston, IL: Northwestern University Press.
- [25] Strauss, A.L. and J. Corbin, *Basics of Qualitative Research - techniques and procedures for developing grounded theory*. 2nd ed. 1998, Thousand Oaks: Sage Publications, Inc.
- [26] Rubin, H.J. and S.I. Rubin, *Qualitative Interviewing: the art of hearing data*. 1995, Thousand Oaks: Sage Publications Inc.
- [27] Radice, R.A., et al., A programming process study., in *IBM Systems Journal*. 1999, IBM Corporation/IBM Journals. p. 297.
- [28] Ciborra, C.U., *The Labyrinths of Information: Challenging the Wisdom of Systems*. 2002, Oxford: Oxford University Press.
- [29] Mintzberg, h. and F. Westley, Decision Making: It's Not What You Think., in *MIT Sloan Management Review*. 2001, Sloan Management Review. p. 89.
- [30] Bach, J., Enough About Process: What We Need are Heroes. *IEEE Software*, 1994a. 12(2).
- [31] Curtis, B., H. Krasner, and N. Iscoe, A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, 1988. 31(11).
- [32] Cusumano, M.A. and D.B. Yoffie, Software development on Internet time. *Computer*, 1999. 32(10): p. 60-69.
- [33] Paulk, M.C., et al., *The Capability Maturity Model for Software*. Technical Report: 93-TR-024, 1993.
- [34] Nandhakumar, J. and D.E. Avison, *The Fiction Methodological Development, a Field Study of Information Systems*. *Information Technology and People*, 1999. 12(2).
- [35] Cusumano, M., et al., *Software Development Worldwide: The State of the Practice*. *IEEE Software*, 2003. November/December: p. 28-34.
- [36] Swartout, W. and M. van Lent, Making a game of system design. *Communications of the ACM*, 2003. 46(7): p. 32--39.
- [37] Baskerville, R., et al., Is "Internet-speed" software development different? *Software*, IEEE, 2003. 20(6): p. 70-77.
- [38] Grazia, S.d., *Time and Work*. in *The Future of Time*, ed. H. Yaker, H. Osmond, and F. Cheek. 1972, New York: Anchor Books.
- [39] Li, E.Y., H.G. Chen, and T.S. Lee, Software process management of top companies in Taiwan: a comparative study. *Total Quality Management*, 2002. 13(5): p. 701-713.
- [40] Starkey, K., *Time and Work: a Psychological Perspective*. *Time, Work and Organization*, ed. P. Blyton, et al. 1989, London: Routledge
- [41] Cusumano, M.A. and D.B. Yoffie, Software development on Internet time. *Computer*, 1999. 32(10): p. 60-69.
- [42] Jacques, E., *The Form of Time*. 1982, London: Heinemann.
- [43] Giddens, A., *The Giddens Reader*, ed. P. Cassell. 1993, London: Macmillan.
- [44] Ward, M., Virtual gaming worlds overtake Namibia, in *BBC News Online*. 2004.
- [45] Castronova, E., *Virtual Worlds: A First-Hand Account of Market and Society on the Cyberian Frontier*. *CESifo Working Paper Series No. 618.*, 2001.
- [46] Gapper, J., Hollywood's game plan has a fault, in *Financial Times - Online Edition*. 2004.
- [47] Nuttall, C., Why has Lara Croft, icon of Cool Britannia, emigrated to the US?, *Financial Times*, 31st May 2004, London.
- [48] Games Developers Conference, March 22-26 2003, San Jose, California.
- [49] Freeman, D., *The Virtual Actor*, EIGF Conference, August 13 2004, Edinburgh.