CREATE_THE GAME PROCESS UPGRADING SERIES
FLESHING OUT

Prototype Workbook 7

The challenge of

IMPROVING MEASUREMENT AND COMMUNICATION OF PROJECT PROGRESS

University of Brighton

CENTRIM

CREATE_THE GAME PROCESS UPGRADING SERIES

Improving measurement and communication of project progress

About the workbook series

The business of video game development has changed significantly in recent years. The complexity and scale of operations by the studios have skyrocketed with the transition to next-generation hardware and so has the investment required to make a game. This has resulted in sector consolidation and the implementation of concept approval and production processes that can squeeze out innovation in gameplay and style and make the work of the developers overly routinised and boring.

These pressures threaten the creativity that has contributed to the success of the UK development sector.

This series of workbooks is the output of a two-year project carried out in collaboration between the Centre for Research in Innovation Management (CENTRIM) at the University of Brighton and 8 world-class UK developers. It is designed to be one pillar of a structure to help games developers retain their creativity whilst producing games on time, on budget and with teams that remain happily challenged in their work.

Organisation of the series

Each of the workbook modules in the series focuses on a particular challenge in video game development. We classify these challenges inside three areas:

CREATING: Processes through which new ideas are generated, combined, tested and refined.

FLESHING OUT: Organisational structures and development processes through which ideas are implemented resulting in a high quality product.

SENSING: Interactions with external parties (ranging from publishers to communities of users or external contractors) supporting the development of a game.

Engaging with this workbook

Who is this workbook for?

The workbook series has been designed primarily for studio managers, producers, and discipline leads. Why? Because these are the people more likely to have a broad view of process and the dependencies between different disciplines. Diffusion of good practice is part of their role. Nevertheless, each workbook contains useful knowledge for anyone facing the topic of the challenge in his or her day-to-day work. If, for instance, you are constantly interacting with external art contractors then the outsourcing workbook will help.

Also bear in mind that implementation of good practice will generate new data and knowledge that can be fed back into the process-upgrading cycle in order to further refine and improve practices in the studio. The workbooks are designed primarily for individuals with a managerial role, but if their potential is to be realised collective engagement with process development across the studio is best.

How do we use the workbook?

This workbook is designed to help you deal with *one* clearly defined challenge. It is an aid for the identification of the challenge's causes, good practices to deal with them and the context of good practice application and implementation.

We organise thinking for improvement around three key questions:

Why does a challenge happen?

In answering this question we enhance our understanding of the context of the challenge.

What can be done to address it?

In answering this question we select good practices targeting the causes of the challenge

How can it be implemented?

In answering this question we align good practice with other systems, processes and protocols in the studio.

For each of these questions we provide options derived from our research. We also leave space for you, the user, to come up with your own options through three different activities.

For best results you should try to assemble a group of people to work together on the activity in a workshop or meeting. It is precisely those working 'on the ground' who might well have information that is more relevant to the solution of the problem at hand. They may also be better placed to highlight the limitations or potential negative side-effects of proposed good practices.

The benefits of this approach include: achieving buy-in by all disciplines at all levels; the high probability of valuable new knowledge being generated on which to base decisions; multi-disciplinary team building is enhanced.

There is often the feeling, however, that getting groups together is more trouble than it is worth and there never seems to be a time when everyone is available. But it is hugely valuable in the end to try as the learning effects can be profound.

This is a prototype 'vertical slice' of the workbook series. We present one challenge with guidance and activities.

For a list of tools and techniques for collaborative problem solving you might want to try, please refer to the **Toolset Workbook.** You can also find some information about the tools at:

http://www.managing-innovation.com/innovation/cda/toolbox.php

http://www.mindtools.com/

The British Government has made available some more strategic tools which can be accessed from the Department for Innovation, Universities and Skills website (http://www.dius.gov.uk/policy/innovation.html).

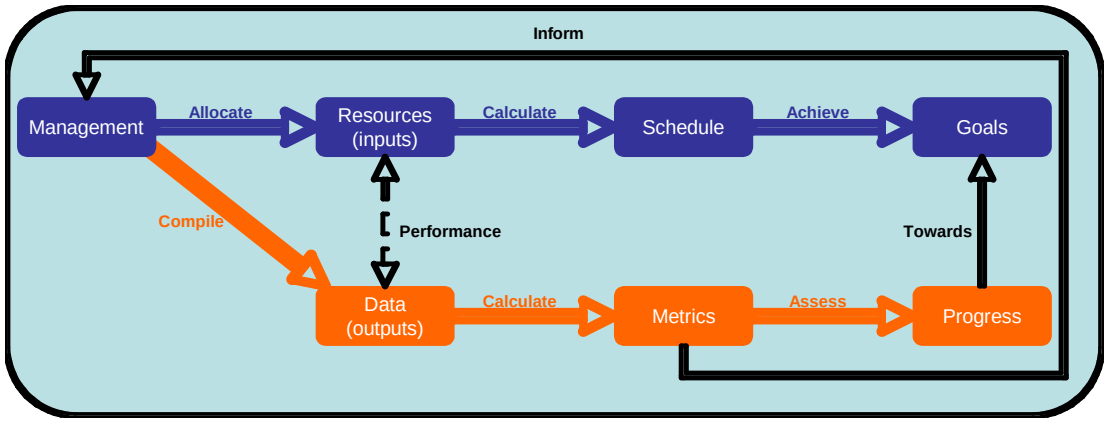
The challenge of improving measurement and communication of problem progress

***Measuring progress in innovative projects* is a particularly thorny issue**. Establishing metrics takes time and thought - resources some might think are better put to use getting that game off the ground or finishing it. Thus, measuring activities often end up relegated to a low priority. However, measuring can enable everyone to judge the state of the project and act upon that knowledge by, for example, re-organising a schedule, redeploying resources for a short while to focus on specific features, or redefining the scope of the game (see figure 1): **measuring can help you make better-informed decisions, and communicate them to your team more effectively.**

What do we mean by metrics? Metrics are a set of indicators for quantitatively measuring and assessing process, actions or events. They enable comparability between individuals, teams, projects and studios. Systematic evaluation can inform daily scheduling and strategy and enhance adaptability by flagging up emerging problems earlier.

Metrics should endeavour to reflect accurately where the game is at, approximating how close are different components to being finished, and what their quality is, as well as the productivity of different teams. They measure the way in which the inputs of your development effort contribute to the output; namely, productivity. Establishing whose time has been allotted to what tasks and whether those tasks have been signed off is a first step towards visualising project progress, but this is only half of the story: you also need to gather data on what is the impact of all of these **inputs** on project progress, analyse them and make the results visible to the team. Being able to see the way in which day-to-day efforts make a difference in the game you are developing is a distinct morale booster and it also makes it easier to justify hard decisions such as dropping a feature that there is insufficient time to deliver in a polished state.

Figure 1: Metrics help ascertain whether the project is advancing according to plan and whether it is necessary to reallocate resources.



CREATE_THE GAME PROCESS UPGRADING SERIES

Improving measurement and communication of project progress

*1- **WHY** IS MEASURING AND VISUALISING PROJECT PROGRESS SUCH A CHALLENGE?*

In this section we begin by suggesting three answers to this question that have come out from our research. This should help you understand the context of the challenge before you start providing your own answers.

1. No agreement on what project progress means

Metrics should tell you how close you are to achieving the ultimate goals of your project; in order to define them it is necessary to make the goals explicit.

Of course you want to make a great game, but what does this mean? Different disciplines might define quality in different ways after all. For example, artists want something that looks striking, designers are more interested in exciting gameplay, and testers want to get rid of all the bugs. And then you want to deliver on time and budget, with as little crunch as possible. So, a key problem is to decide whether all these things can be achieved at the same time, how important they are in respect to each other and what the priorities are for the team.

2. We cannot establish the right metrics for progress

Even if you have established what you want to accomplish you need to define metrics that measure your progress towards those goals, collect data to calculate them and then interpret them.

Resource allocations (see figure 1) are often used to approximate progress in different areas of the game. For example, if one of your goals is delivering ‘super innovative feature X’ (that will differentiate your game from all the competition) the labour-hours scheduled for its implementation are used in order to measure progress in that area. The problem with doing this is that you are looking at inputs of the development progress, rather than outputs (what you are coming up with at the end of the development cycle). Your metrics should not just be telling you that there are 10 developers working on super-innovative feature X, but more importantly, whether their effort is contributing to getting super-innovative feature X done on time. This knowledge makes it possible for you to assess whether it is necessary to get more developers working in the feature; for example, if it is running behind schedule, it is ‘buggy’ or not sufficiently polished.

Task sign-off is another common measure of progress. The thinking goes like this: “*the lead or line manager responsible for an area of the game assessed a task carried out by a developer, and decided that it was done. All the tasks are getting signed off according to plan, so this means we are getting close to finishing the game*”. Again this has problems because translating signed-off tasks into game quality is not as straightforward as we would hope. There is a big jump between ticks on a spreadsheet and a fun-to-play, polished game experience.

3. We do not have a culture of measurement

In order to elaborate metrics you need relevant and timely data, which need to be collected and processed. This can be an arduous effort, both from the point of view of process (how do you get your developers to submit data that can be used to elaborate metrics?) and studio culture. Regarding the latter issue, there is the problem of developer mistrust towards 'formal management procedures' which can be perceived as creativity-stifling.

SUMMARY: **WHY IS MEASURING AND VISUALISING PROJECT PROGRESS SUCH A CHALLENGE?**
1. We don't agree on what we mean by project progress
2. We can't establish the right metrics for progress
3. We don't have a culture of measurement

ACTIVITY 1- IT IS YOUR TURN TO UNCOVER SOME MORE WHYS ***Ask yourself, and other members of your studio, why do we have so much trouble measuring and communicating project progress?***

2- WHAT GOOD PRACTICES CAN WE PUT IN PLACE IN ORDER TO ADDRESS BARRIERS TO MEASURING?

For every **WHY** there are **WHATS**. For every challenge there are practices that can be introduced to deal with them. In this section we present good practices that can help alleviate or eliminate the challenges described above. There are also two activities: first we suggest another WHY, a barrier to efficient measuring and communication of project progress for which you should come up with some pertinent good practices. Then you are asked to identify good practices addressing the WHYS you came up with in Activity 1 above.

Don't forget that identifying a good practice is only the first step towards solving a problem: good practices need to be implemented and can create their own challenges. We focus on these issues in Section 3 when we ask another key question, HOW do we make good practices work?

But before that let us throw a few good practices your way:

1. WHAT can you do to establish a clear definition of project progress?

a. Make goals explicit and visible

Once you know what your goals are you can start assessing your progress towards achieving them. It is particularly important to ensure that there is a studio-wide awareness of these goals, so it is a good idea to give them high visibility perhaps using props such as large boards and posters.

b. Quantify goals whenever possible

In a perfect world, it would be possible to quantify all your goals so you could assess how close you are to attaining them as the project advances. This is very difficult in the case of video game development, where many of the goals are aesthetic (does it look nice?) or experiential (is it fun to play?); two highly subjective variables. This does not mean that it is not worth setting some quantitative targets in specific areas where it is easier (for example, frame-rate per second or bug count at different milestones), against which you can benchmark the evolution of the project.

2. WHAT can you do to establish relevant metrics of progress?

a. Deliver functionality instead of components

As we said, inputs are an imperfect measure of project progress and so are some outputs such as signed off tasks, which neglect the systemic nature of video games (the whole of the game is bigger than the sum of its features). It seems that the best way of measuring where 'the game is at' is by actually playing it, something that can only be achieved by focusing the development cycle on delivering game functionality; i.e. something that you can play. This replaces *waterfall* methods where components built during development are integrated at a late stage of the project when there might be too little time to react if something goes wrong, or the game is found to be broken (see figure 2).

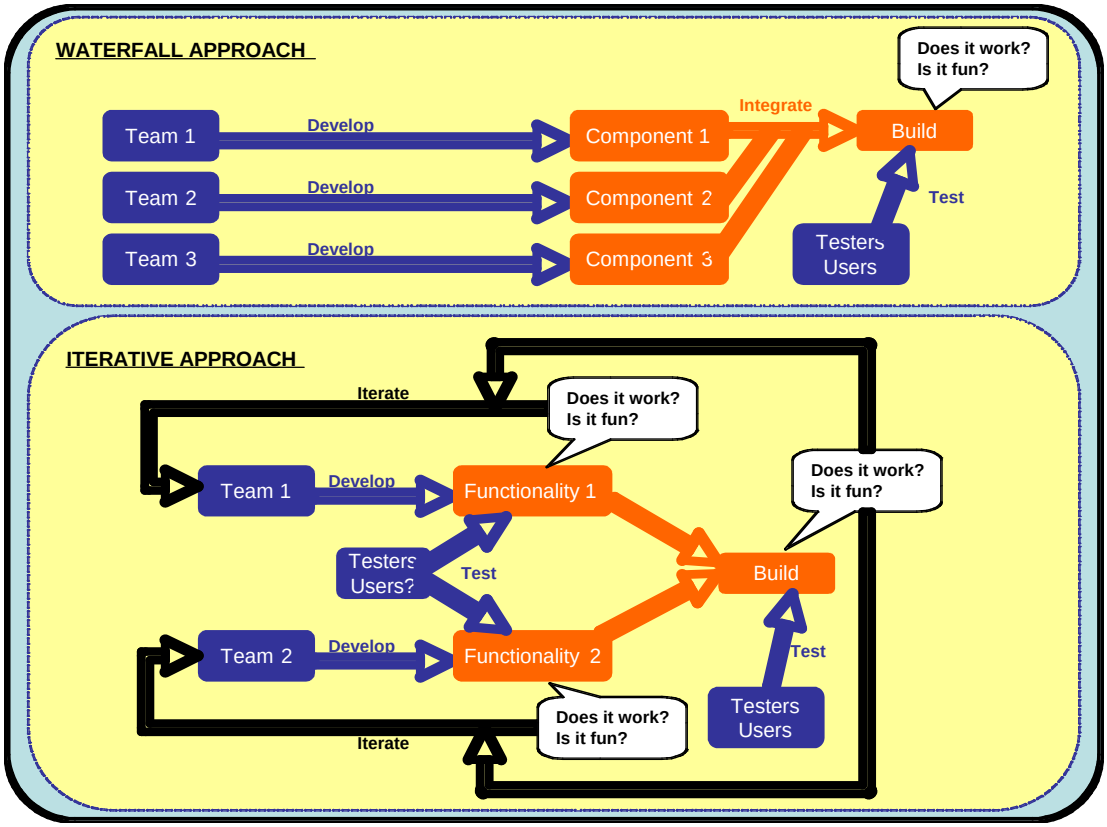
This also gives testers something to start working on from an early stage and improves studio morale by highlighting how everyone's day-to-day efforts contribute to the evolution of the game.

b. Internal QA cycles

Another good way of keeping track of the evolution of the project is through the implementation of internal QA and testing cycles. This includes automated testing suites to ensure that daily builds don't break, informal play-testing by developers or even blind-testing by users in order to balance the game as it is produced.

All of these strategies make it easier to flag up and address serious problems with the game, rather than leaving them for a phase of protracted testing at the end (see Figure 2).

Figure 2: Delivering functionality in shorter cycles than components at the end helps to assess earlier where the game is at.



c. Studio walkarounds

Studio walkarounds (where management and leads visit different teams and departments and developers show them what they have been working on recently) are part of a more informal strategy to keep track of the project progress. These walkarounds make it possible to gather intelligence about the state of the game in different areas, flag dependencies and improve communication between managers and teams.

3. WHAT can you do to promote a culture of measurement?

a. Automated data collection

It is a good idea to reduce the opportunity cost of collecting metrics by implementing protocols for the automatic collection of data, to be displayed in a user-friendly way where possible. Some project management tools have such features, enabling you to establish who has done what, who

gave the sign-off and how close you are to achieving specific milestones. Another area where automated data collection can be very useful is during user testing and debugging. In the case of the former, data on user behaviour and performance is a good complement to potentially biased responses in questionnaires and interviews. In the latter, accessing bug numbers and severity will enable you to determine project progress during post-production.

b. Project management training for managers and leads

It is often the case that project managers are former developers promoted to a management position in a studio because of their leadership or good communication skills. Although these are of course the essential characteristics of a good project manager, some formal project management training will improve their understanding of established ways of scheduling, common metrics and tools for visualising project progress.

This good practice could also be applied to those discipline leads more highly involved in management duties.

SUMMARY:

<u>WHAT</u> GOOD PRACTICES CAN WE PUT IN PLACE IN ORDER TO ADDRESS THE MEASURING CHALLENGE?

WHY IS IT A CHALLENGE?	APPROPRIATE GOOD PRACTICES
1. We don't agree on what we mean by project progress	1. Make goals explicit and visible 2. Quantify goals whenever possible
3. We can't establish the right metrics for progress	1. Deliver functionality instead of components 2. Internal QA Cycles 3. Studio Walkarounds
4. We don't have a culture of measurement	1. Automated Data Collection 2. Project Management training for managers and leads

ACTIVITY 2- IT IS YOUR TURN TO SUGGEST SOME GOOD PRACTICES ***Here you have another possible WHY that explains the difficulties of measuring project progress. Can you think of any good practices that could be set up in order to address this WHY?***	
WHY IS IT A CHALLENGE?	**APPROPRIATE GOOD PRACTICES**
o **It is difficult to measure the actual productivity of our developers because we don't know how much time they spend working on the tasks they have been assigned.**	

ACTIVITY 3- IT IS YOUR TURN TO SUGGEST SOME GOOD PRACTICES ***Now can you think of any good practices to address the WHYS you came up with in ACTIVITY 1?***	
WHY IS IT A CHALLENGE?	**APPROPRIATE GOOD PRACTICES**

3- HOW CAN WE MAKE THESE GOOD PRACTICES WORK?

The devil is in the detail: good practices have limitations and also unintended impacts that need to be taken into account. When applying them you need to think about the 'bigger picture'; that is, the ecosystem of practices and routines of the studio.

You should think about whether a new practice fits or clashes with the way people work, in order to avoid ineffective or even detrimental implementation (when, for example, practices clash with each other). You might even decide that, in order to reap the full benefits of a practice you need to change more things around the studio, this is the way organisational transformation begins!

You also need to be aware of how different teams inside your studio develop components later to be integrated into the bigger whole and the trade-offs that may emerge between their efforts (for example, when there are not enough resources to achieve all that you wanted to put in the game and you have to decide what features to drop, creating disappointment in a team).

This is one of the reasons why it is particularly important to get the broader studio involved in the discussion about good practice implementation. The techniques presented in the Toolset Workbook should help you in this collective effort.

We support this deliberative process now by focusing on two questions, the answers to which should help make sense of the HOWS above:

a. What are the limitations and implementation details (what we call '**the small print**') of this practice?

b. What other areas of good practice is it linked to (what we call the '**dependencies**')?

Regarding the dependencies, we think about good practices systemically, and acknowledge that many of them are interrelated. So, when necessary, we refer you to other workbooks containing information that should be pertinent if you decide to implement the good practices presented here.

But perhaps the best way to explain it is by giving some examples linked to the best practices we have suggested above. We also provide you with a couple of activities you can use to discover the 'secrets of implementation' of the practices you came up with in Activities 2 and 3.

1. Good practices to establish a clear definition of project progress

a. Make goals explicit and visible

HOW do we make this work?

The small print

Do it early

The pre-production phase can make or break a game. Defining the goals early, establishing them at the right level of detail and achieving a collective understanding of priorities (and what happens to low priority assets, features, etc.) are important attributes.

Be ambitious, but realistic

Some people say that setting ambitious goals gets the team motivated and helps avoiding mediocrity. It can also be a recipe for lots of crunching at the end of the project. We cannot help you make these decisions, but make sure you are aware of the trade offs and potential consequences of the scale and scope of your goals.

Dependencies: **Concept workbook, prototyping workbook**

Get people involved

Involvement is an important way of getting people to be open and collegiate, and create a feeling of ownership and excitement about the project. Idea generation workshops across a studio can be effective in this respect. Get everyone together to discuss what the game is about, and perhaps suggest ideas for features. This is also a really good way of flagging up potential problems and bottlenecks as soon as possible.

Dependency: **Idea generation workbook**

Create visible benchmarks

Sometimes it can be difficult to describe project goals in a piece of paper. Yes, you want to make a thrilling or stimulating game, but you need to show what you mean by that. The same applies to, for example, the visual style. A good idea might be to create prototypes, 'vertical slices' and reference artefacts (e.g. concept art, mood boards) to show everyone what the goals of the project are in a language that is easier to understand. This also makes it possible to see which gameplay mechanics have early potential.

Dependencies: **Prototyping workbook, Documentation workbook.**

But also document the goals

It is a great idea to cover the walls of your studio with visible props that remind people of the shared vision for the game, but for some other goals it might be necessary to get into further detail; that is, what formal documentation is for.

Dependencies: **Documentation workbook.**

And stay flexible

Of course, projects never run according to early plans and rigid adherence to them can have negative impacts. These explicit and visible goals will probably need to evolve as development unfolds. The whole 'system' that supports the development process needs to be adaptable or flexible. And above all, you need to make sure that any changes in priorities are well communicated and, if necessary, explained to your team.

Dependencies: **Evolving the game workbook**

b. Quantify goals whenever possible

HOW do we make this work?

The small print

Leverage your developers' expertise

The best people to obtain reliable data on 'how long is it going to take' and 'whether it is

possible' are the developers on the ground. That said, different disciplines respond in different ways – some are naturally conservative, others overly optimistic. Some are subject to crunch, others are not. Understanding these characteristics is a skill learned by managers, leads and producers.

Dependencies: **Scheduling workbook, Communication workbook, Team workbook, Middleware and tools workbook**

Use past experience to support current decisions

Having some historical data about your team's behaviour supports the quantification of goals by for example telling you that a certain developer always tends to be overtly optimistic when estimating the time it will take him or her to carry out a task, or what can be achieved: this is another reason why measuring is a good idea not just in the context of your current project, but also thinking about the future.

Dependencies: **Scheduling workbook**

2. Good practices to measure project progress more clearly

a. Deliver functionality instead of components

HOW do we make this work?

The small print

Establishing the level of functionality

It is all well and good to say that your teams are going to develop functionality instead of components. Now you need to define what functionality is. In the first instance one might see it as playability – can it be played, however imperfectly? In order to achieve that functionality, do you need to integrate different features, or can some features be played on their own? If you need to integrate features, how do you synchronise the activities of different teams so that they create outputs that work together?

And some features (for example, systemic or architectural features) might well simply have to be implemented at the end of the project.

Dependencies: **Development Cycle Workbook**

Team composition

If you decide to move towards a system focused on the delivery of functionality you might have to change a few things around the studio. Team composition is a key one – you will probably need to organise development around interdisciplinary teams, and this might be an important challenge from the point of view of governance.

What we mean by this is the following: when you have functional teams (e.g. artists, programmers, designers) working on components of the game it is quite clear who is in charge of each area – the lead programmer manages the coders, the lead artist deals with the artists, etc. On the other hand, with interdisciplinary teams you might end up with team leaders on the one hand and disciplinary leads on the other. Deciding who calls the shots in blurry areas can become a source of tension.

Dependencies: **Development Cycle Workbook, Management and Governance Workbook, Communication Workbook.**

b. Internal QA cycles

HOW do we make this work?

The small print

Testing requires resources

Time needs to be set aside by the QA team to establish the testing cycles – coding and analysis, as well as feedback loops between testers, developers and management. QA tends to be a weak link for studios with many being reliant on external sources. Evidence from our project suggests that development teams in studios with internal QA teams benefit from their close and rapid feedback.

Get people outside the team to test the game

Best results may come from developers not directly engaged in the project, because they do not know what to expect, might be less tolerant with errors, and also come up with unexpected ideas.

Dependencies: **Testing Workbook**

c. Studio walkarounds

HOW do we make this work?

The small print

Constructive management

One hallmark of good management is visibility and taking an interest in the work employees over and above the deadlines. Time needs to be set aside for informal interactions, and they should be approached constructively. It is not about catching people out, it is more about sensing mood, morale, incipient problems, etc.

Dependencies: **Management and Governance workbook, Team Workbook**

Establish clearly expectations about polish

Managers also need to be aware that the link between effort and polish, for example, may be blurred. A lack of polish does not necessarily mean that people are not doing their jobs. What is more, if the game is being developed with an iterative approach, excessive polish early on might be detrimental and wasteful.

Dependencies: **Communication Workbook, Development Cycle Workbook.**

3. WHAT can be done to create a culture of measurement?

- ## Automated data collection

HOW do we make this work?

The small print

Get the relevant data and store it

Managers need to be clear about what data are being collected and why. Collecting data for the sake of it is inefficient. It can also alienate the team and create an unnecessary information overload. One way around this is to automate data collection within scheduling tools; this also secures simple comparability between plans and reality. Bespoke tools are sometimes required for this. One size does not fit all.

You also need to store the data you collect safely. As we said before, it can be useful when you set the goals and schedules in the next project

Dependencies: **Middleware and tools Workbook, Scheduling Workbook**

Show developers that measuring is for the good of the project

You might need to demonstrate to developers that metrics are not collected because of trust issues, or in order to routinise their jobs, but to help them get on with their work and improve the outcomes of the project. You can achieve this through clear, open and honest communication. Good workshops and showing the utility of interventions supported by metrics (through charts, etc.) help in this respect.

Dependencies: **Team Workbook, Management and Governance Workbook**

- ## Project management training for managers and leads

HOW do we make this work?

The small print

External management training has an opportunity cost

Training is always time-consuming and on the face of it, it can feel like a waste. Many studios prefer to have all of their training on-the-job. However, whilst this may be appropriate for core development activities (few studios seem to have any confidence in external training providers)

more generic activities such as project management can effectively be provided externally and the cross-sector learning may be a source of good ideas.

Dependencies: **Learning Workbook**

Not all project management techniques are useful

Some project management tools and frameworks might be difficult to apply in the rapid moving, highly innovative context of video game development. Some examples of this include process and document driven, formalised techniques such as the Capability Maturity Model or the waterfall methods mentioned before. You need to assess the extent to which the different programmes and techniques available are relevant and useful to your studio.

Dependencies: **Development cycle Workbook**

Choose the right people for a managerial position

There remains a tendency rapidly to promote people to managerial positions from within studios. Rapid promotion without adequate management training can lead to problems as people are moved beyond their capabilities, or into areas where their specialist skill sets are wasted. If you promote an excellent programmer to a managerial position you may end with a programmer less and an inadequate manager. Criteria need to be established for promotion such as communication, financial and project management skills and training.

Dependencies: **Management and Governance Workbook, Team Workbook**

SUMMARY:

***HOW* CAN WE MAKE THESE GOOD PRACTICES WORK?**

WHY IS IT A CHALLENGE?	APPROPRIATE GOOD PRACTICES	SMALL PRINT
We don't agree on what we mean by project progress	**Make goals explicit and visible**	▪ *Be ambitious, but realistic* ▪ *Do it early* ▪ *Get people involved* ▪ *Create visible benchmarks* ▪ *But also document the goals* ▪ *And stay flexible*
	Quantify goals whenever possible	▪ *Leverage your developers' expertise* ▪ *Use past experience to support current decisions*
We can't establish the right metrics for progress	**Deliver functionality instead of components**	▪ *Establishing the level of functionality* ▪ *Team composition*
	Internal QA Cycles	▪ *Testing requires resources* ▪ *Get people outside the team to test the game*
	Studio Walkarounds	▪ *Establish clearly expectations about polish* ▪ *Constructive management*
We don't have a culture of measurement	**Automated Data Collection**	▪ *Get the relevant data* ▪ *Show developers that measuring is for the good of the project*
	Project Management training for managers and leads	▪ *External management training has an opportunity cost* ▪ *Not all project management techniques are relevant* ▪ *Choose the right people for a managerial position*

ACTIVITY 4 - IDENTIFY SIDE EFFECTS, CONNECTIONS AND IMPLEMENTATION ISSUES IN THE GOOD PRACTICES FROM ACTIVITIES 2 AND 3		
It is difficult to measure the actual productivity of our developers because we don't know how much time they spend working on the tasks they have been assigned		

4- WHAT we have learned

In all industrial sectors projects are inherently unpredictable and liable to slippage. In projects such as videogames where creative people from a range of disciplines interact, the unpredictability is increased. The sector does not have a culture of measurement. But in the era of next-gen development, balancing creativity and productivity is a key challenge. The necessary metrics are sometimes not obvious. It may not be about numbers exclusively, qualitative measures are required and the ability to collect, analyse, present and act upon data is a skill to be embraced. This workbook provides some practical steps for studios to meet the challenges.