Production 101

Ruth Tomandl
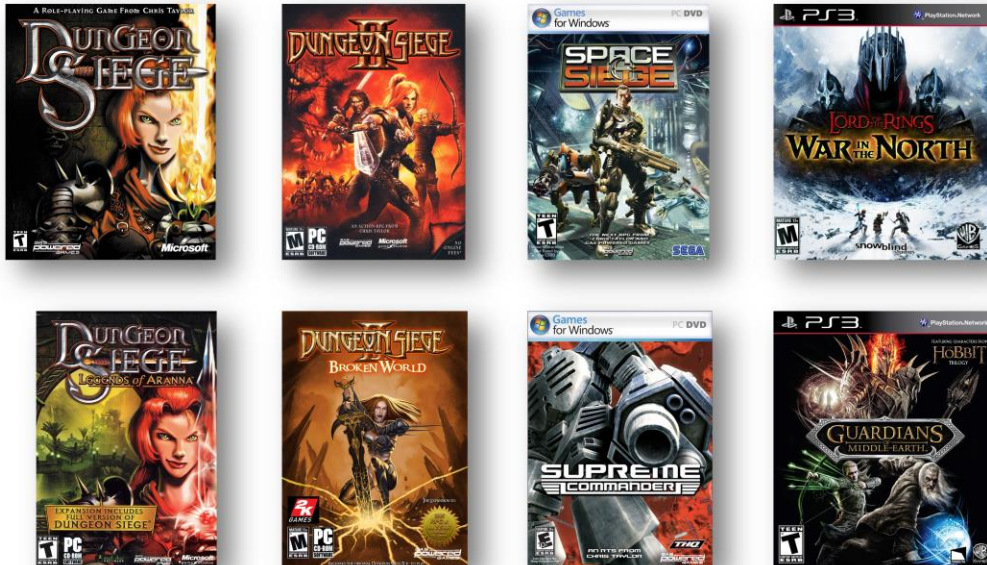Sr. Producer, Monolith Productions

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 25-29, 2013
EXPO DATES: MARCH 27-29
2013

This talk is for the Producer Bootcamp [213] at GDC 2013.
The description is on this page:
http://www.gdconf.com/conference/tutorials.html

The following notes include powerpoint slide text (black text)
and speaking notes (*blue italic text*).

The talk is intended to be 45 minutes, with 15 minutes of Q&A
afterwards. The target audience for the talk is people who are
currently in the games industry and want to go into
production, or people who have done so recently and have
been producers for less than a year.

Introduce self, my background: Hello, my name is Ruth Tomandl and I'm currently a Sr. Producer at Monolith Productions. I've been a producer for 5 years, and before that I was a level designer for 7 years. I've learned a lot during the transition from content creator to producer and during my time as a producer, and I've been lucky to have very good leads and mentors along the way, so I want to share some of the things I learned to help all of you on your career path.

Before I start, who in the audience is currently a producer, for how long, who wants to be a producer, etc.

This talk will focus on development team production, and specifically how to use your development background effectively in a production role.

2

Agenda

1. Overview of Production
2. Play to your Strengths
3. Fix your Weaknesses

In this talk, I'm going to give an overview of what production is and how it's likely different from your former (or current) role, discuss how to play to your strengths as you transition to producer, and talk about how to avoid some common pitfalls, including any weaknesses you might have from your former (or current) role.

I'm going to save 15 minutes for questions at the end of the talk, so please note anything you'd like more detail on. This is a lot of ground to cover so it will be pretty broad.

1. Overview of Production

Production is making sure the game gets made.

Shall we finish our game?

Production is making sure the game gets made.

Making sure content and decisions get made, but not making either yourself.

You will be presenting decisions to others, and the information to make them, and making sure the decisions get made, but not actually making the decision yourself.

This is difficult to get used to and some people have a hard time with the separation.
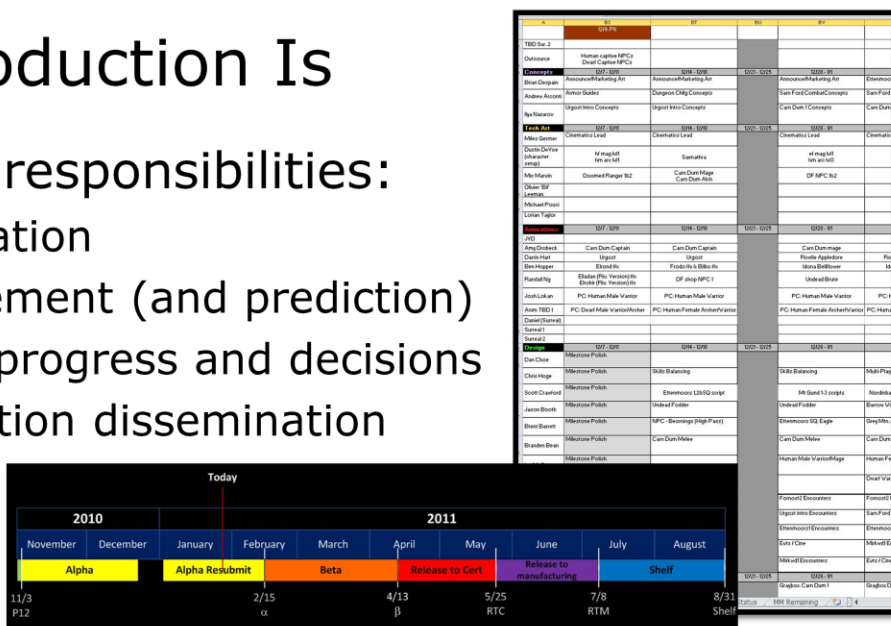
Types of Producers

Other types:
- Publishing producer
- Studio head
- Production coordinator
- Development director
- Production manager

This is my current development team for a 120-person team. Our executive producer oversees the whole team and communicates with the publisher and executives. He looks for problems affecting the entire team or the whole schedule. The two senior producers, me and Nate, oversee art and design respectively: each team is about 30-40 people. We make sure communication between our teams and the rest of the project is good, and look for blockers/communications breakdowns and fix them. We drive decisions getting made that affect our teams. Producers and associate producers oversee smaller sub-teams in more detail. They track tasks for individual team members. The Content director shown here isn't a producer, but he does a lot of tasking and communication work.

What Production Is

Production responsibilities:

- Prioritization
- Measurement (and prediction)
- Driving progress and decisions
- Information dissemination

Production responsibilities:

Prioritization

> Requires knowledge of the overall project priorities and of what is needed from your team.

Measurement (and prediction)

> Requires record keeping.

Facilitation (unblocking)

> Requires boldness and knowledge (of what's blocking your people and how you can get it fixed)

Information dissemination

> Requires knowing everything possible about the state of the project so you can get the right information to the right people. Probably the thing you'll spend the most time doing.

"Doing the schedules" is the first two.

The hardest part of this for me is measurement. Setting up a process is easy. Maintaining it, and measuring progress
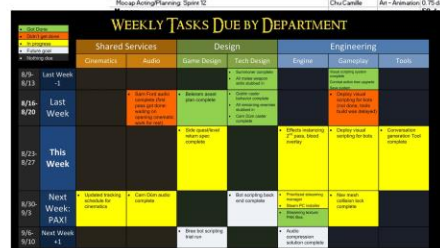
against it, is hard. Every team I've worked on has had at least one whiteboard full of abandoned post-its, or task board on the intranet that hasn't been updated for 18 months. Keeping those processes useful through project changes, feature changes, estimates that were off, or being sick for a week is extremely difficult.

> On one project, a producer had worked with the level design team to set up scrum, but had stopped maintaining the scrum board or having daily stand-ups. However, because the designers had been told to put a post-it note in the 'for review' column of the board whenever they finished something, there were piles of post-its on the board, on the floor around the board, that nobody was looking at. The designers continued to update the board but they clearly didn't believe in the process. They'd smack another post-it in the "for review" section, watch a bunch more fall on the floor, sigh, and then walk away. When devs say they hate production and think that process is useless, this type of thing is usually the reason why.

Information dissemination is also hard and time consuming. No matter how many people you give a piece of information to, there will be someone you forgot who needs to know it. This is particularly difficult on projects with high rates of change and projects with large or distributed teams: the more systems you can get in place for this (review meetings, intranets, documentation, mailing lists) the better.

What Production Is

Other possible responsibilities:

- Requirements gathering
- Running meetings, taking notes
- Facilitation (unblocking)
- Conflict resolution
- Bug triage and assignment
- Being a force for positivity

Other possible responsibilities:

Requirements gathering and checking

> Making sure your team knows exactly what they should be doing

Running meetings, taking notes

Documentation

> These three mostly fall under information dissemination

Validation

> confirming to someone that the thing they were already going to do is the right thing to do. If people trust you, they will ask you for this a lot.

Bug triage and assignment

> Part of prioritization

Being a force for positivity, encouragement

> "Team cheerleading". This is often more of a leadership skill than a production one, but producers are often looked to as leaders by the team.

I've done spokesperson work, and research for the Tolkien

games.

Random tasks that nobody wants to do: i.e. anything that isn't programming, art, design, qa, or audio.
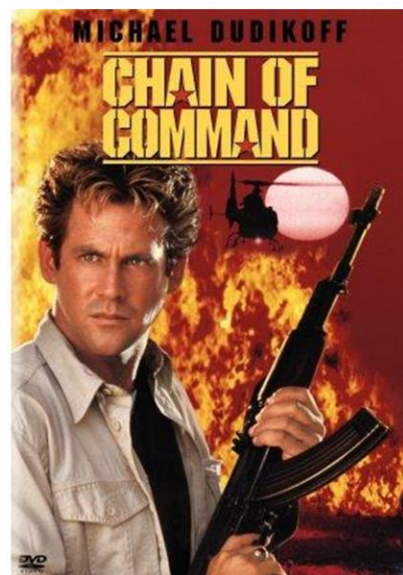
What Production Is

Be useful.

Do whatever it takes to get the game done. This is me setting up a laptop at a Lord of the Rings convention in Germany to let fans sign up for our newsletter. If that's what it takes, jump in and do it.

What Production Isn't

- Directly working on the game
- Deciding what the game is
- Art direction
  - Or direction of any kind
    - For the love of god, don't direct anything.

Directly working on the game.

> You may still work directly on the game, but not under your production role. Sometimes (especially on smaller teams) people have 2 or more roles, and it's good to understand them and be able to separate them.

> Not directly working on the game can be a hard transition for people who are used to working on the game

> > It was difficult for me: I went from being a level design lead, owning the levels, jumping in and fixing bugs, on an engine I had 7 years of experience working in, to scheduling the artists on a game and engine I was completely unfamiliar with. I felt helpless at first, like I couldn't change or fix anything, and that made me want to tell people exactly what to do in a very hands-on way, which caused conflicts with the leads.

> > It was also extremely difficult for me to tell whether I was doing a good job, since I wasn't doing work that other devs could easily evaluate.

Fortunately, I had a good lead who gave me feedback regularly but it was still difficult to get to a point where I felt effective.

I should note that this is a point of contention: some teams have a culture of "everyone jump in and help wherever they can", regardless of role; this works well on small, mature teams but not as well on large or inexperienced teams.

Deciding what the game is

This is the product owner, lead designer, creative director's job.

As a producer, your job is to inform that person of what your team can do, their progress, and recommend strategies. (This is the information you send upward)

Also to keep the team informed of what the game goals are (information sent downward).

You also have to make sure that decisions get made when they need to: driving decisions. You can recommend what you think those decisions should be, but they're not your decisions. They'll feel like they are sometimes, since you are communicating them to so many people, but making those big 'what the game is' decisions is not your job.

Art direction. Or direction of any kind. For the love of god, don't direct anything!

The worst producers I've worked with all have this in common: they act like directors. This is incredibly damaging to the project: it adds chaos because people are hearing different direction from multiple people in authority, and producers aren't generally very good directors (if they were, they'd get hired as a director).

Definitely give opinions upwards, but don't give direction downwards (to your team). To your team, you are an authority figure because you hold the information. Do not disagree out loud with the actual directors in each of those areas. Not only does this hurt the project, but it makes you hard to work with.

This was the first and hardest thing that I had to learn going from design to production. I had design opinions and I knew they were right! I'm ashamed to say that I got in multiple arguments in meetings with the design

director on my first project. It was very damaging; fortunately I had a good manager who told me that it was hurting the project and to knock it off. At the time I didn't even realize I was doing it, since it was such a key part of my former role.

It is valuable to use your experience to help drive decisions; use it to understand problems and present possible solutions and make sure a decision gets made early, but avoid the "I was a designer, so I'm right!" attitude like the plague.

Your Assets as a Producer

- Credibility
    - Understand your team
    - Share information with your team
    - Be realistic with expectations
    - Ask questions
    - Define success for your team
    - Use your previous experience

Your most important asset is credibility. Credibility is the biggest difference between good and bad producers. You can only do your job well if people trust what you say, because so much of your job is getting information to people or communicating priorities.

How do you build credibility?

Understand the jobs of those who report to you

> Understand what's easy for them, what's hard for them, where they face challenges. What do they do all day? Why is that important?

> This can be easier if you're producing people with the same job you had, but watch for differences and make sure you understand them.

Share vision and information with the team

> Make sure the team knows what the vision for the product is, and what they're building.

Be realistic with expectations

> Know what the team can do. Allow the team to exceed

10

your expectations, don't set them so high that they're continuously impossible to meet.

Be clear about what success looks like

If things are going well, be specific about what exactly is going well. People will actually believe you, and they'll be more confident in their work.

Ask questions

Ask questions humbly. Don't be afraid to ask stupid questions. Again, the answer probably isn't obvious to everyone.

There is a time and place for this. If you're the only non-technical person in a meeting of network engineers and you don't know what an online service layer is, ask after the meeting. But if there are other non-engineers in the meeting and they look lost, and it's important for them to not be lost, ask.

Use your previous experience as a guide

Saying "I used to be a level designer" does nothing for your credibility. Knowing that world art needs to be involved in a combat decision because it could mean extra work for them does.

Your Assets as a Producer

- Credibility
- People knowing that you'll make their lives easier
- Your background

2. People knowing that you'll make their lives easier.

(Not nearly as important) is people knowing that you usually make their lives easier, so they come to you when they run into something. This will give you information and let you do more of your job. Be Useful.

How do you do that? Often little things: being dependable at giving information. I don't bother people who have the "working hard, in the zone" look, I try to save up multiple things to bring up all at once, and I try to find something I can do for them in every conversation, even something small. The best feeling for me is when someone who used to look apprehensive when I walked over starts looking up and smiling instead.

I vividly remember the feeling of a producer coming over to my desk and either: bracing myself because I know they're about to waste a bunch of my time, or: being happy because I know they're about to save me some time or give me information I need. So it was always a high priority for me to be the second one.
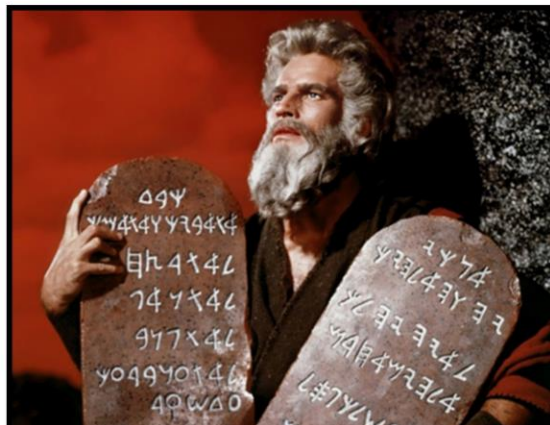
3. Your background

Your background gives you a huge amount of information, and information is your main tool. You need information to prioritize tasks correctly, to watch for problems, to know who to ask about a problem. If you've worked on successful games, you know at least something about what makes a game successful, and if you've worked on failures you know at least something about what makes a game fail. If you've worked in art, you know some things that slow down art tasks, and if you've worked in engineering you know what's hard about getting information from other departments. You will catch problems that people without your background would miss.

2. Play to Your Strengths

What kinds of people make good producers?

- Self-motivated
- Good communicators
  - Especially of bad news
- Good motivators

If you're at this talk, you've already decided that you're interested in production. Why? And what skill sets and experience do you have that will make you a good producer?

Moses was a great producer. He had to carry out someone else's plan (in this story, God is analogous to the creative director) with a team (the Israelites) who didn't really want to go along with it, and he had to deliver a lot of bad news to a lot of people. But he got it done!

Self-motivated

    able to look for (and find) useful work to do

Good communicator

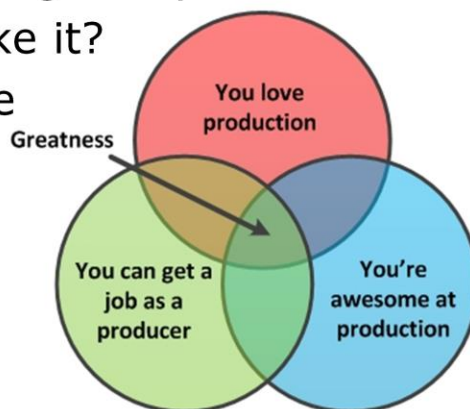   Able to communicate bad news, and able to ask for bad news

Good motivator

    Able to get people to do things they might not want to do

2. Play to Your Strengths

What kinds of people make good producers?

- How do you know you'll like it?
- How do you know you'll be good at it?
- How do you become a producer?

Greatness

You love production

You can get a job as a producer

You're awesome at production

1. Do you love facilitating others' work? Do you love finishing things and moving on? Then you'll probably love being a producer! Artists, people who want to iterate forever, are necessary on a quality team but you need producers driving them to ship to balance that out. Companies that don't have producers often don't have a lot of pressure to ship.

2. Remember the skills you already have, so that you'll be more confident that you can learn the rest.

3. If you're already a developer, work towards doing more production-y tasks like documentation, keeping an eye on your own task lists and priorities, and seeing which of your team members are predictable in their estimates. If you're a student, make sure you are able to work as a producer on your student projects, and get them shipped!

4. If you're doing work you love, and are good at it, things will be great!

Develop Your Skills

- Learn good questions to ask
    - And how to turn bad questions into good ones
- Talk to devs
- Learn what's easy (and hard) for your team
- Police your habits

Learn good questions to ask

> And learn how to turn bad questions into good ones.
>
> "Can we do X?" → "Do we have time to do X?"
>
> "Do we have time to do X?" is a much better question than "Can we do X?" because the answer to "Can we do something?" is always yes. You have a team full of very smart people; nothing is impossible. Know when people are asking you meaningless questions and don't be afraid to rephrase their question and answer the one they should have been asking.
>
> > This might make you feel like kind of a dick, but it's necessary for your credibility.
> >
> > Don't tell people what they want to hear (and they want to hear "yes!"); tell them what they need to know.

Talk to devs

> especially on other teams or in other disciplines.
>
> If you don't talk to devs regularly, see if you can get used to it. Especially if you are intimidated by them;

you'll have to power through it sooner or later, and it may as well be sooner.

Take a dev producer to lunch and pick their brain. We love this because it makes us feel wise and smart.

See if you can go talk to a dev on another team about something you're working on or testing.

Find someone who does stuff that you know nothing about, and ask them what they do. See how much you can learn in 10 minutes.

Learn what's easy (and hard) for your team

And other teams

One important thing you'll have from your non-production background is a general idea of what's easy to do and what's hard to do. This is extremely important but not often discussed. The tendency is for teams to swing for the fences for fear that if they don't, they will be seen as having a lack of ambition or skill. True ambition and skill comes from having an understanding of where the easy wins are and which features will require a lot more effort. Your background will make it easier to be honest with what you can achieve and set up your team for success.

Try to learn this about disciplines you're not familiar with (by asking a lot of questions), and make sure your instincts are right by often voicing your assumptions: "I believe X is pretty easy and low risk, but Y is much harder and riskier, is that right? How much longer do you think Y would take than X?"

> This is one thing that your background might make it harder to learn: be careful about assuming that the disciplines you're not familiar with are similar to the ones you are familiar with. If you come from art, you'll have an artist's perspective of design and engineering, which is different from the perspective you'll need as a producer. It will take effort to recalibrate your difficulty meters for each project, discipline, and honestly for each developer you manage.

Police your habits

Because you probably won't be on a big team of producers that you can learn from and will probably be largely on your own, and because it's more difficult to answer the question

"Am I a good producer?" than the question "Am I a good artist/programmer/tester/designer?", a good lead or a good mentor is incredibly valuable; someone who can give you good feedback.

Ask the leads of the teams you're producing for regular feedback.

Self-analyze; are your day-to-day habits helping the project, the team, and the company? Produce yourself!

Ask for feedback: if people are reluctant to give it to you, regularly ask your team questions like "Are your priorities for the next 2 weeks clear?" "Is there anything else I could do to help you get X done?" "Am I making your life easier or harder? What can I do to do more of that/fix it?"

Good Habits and Bad Habits

The best producers:

- Remain calm
- Ask lots of questions
  - At appropriate times
- Care about their game, their people, and their company
- Are positive but realistic
- Take care of business

The best producers:

Remain calm

> Do not react negatively to bad news. Take it in stride. If you react badly, people will stop coming to you with bad news, and you need that bad news to do your job.

>> I expressed shock and dismay at a feature change and could immediately see the poor designer's face fall and knew that he'd never want to bring bad news to me again.

> Try to stay serene; it's not always easy.

>> Try to have regular 1-on-1s with your manager; if you need to vent, do it here and save your serenity for the floor.

> Make sure to react positively to good news, though!

Ask tons of questions, do a lot of listening

> Are willing to ask questions that don't make them look good

> Are willing to say redundant things

>> As I have been doing, and will continue to do, in

this talk.

I'm also very nosy, which helps me know more but can be a detriment at times. Sometimes I have to remind myself that there are things that really aren't my business.

Care about their project, their people, and their company

Care about your people, but not too much. Your job is to ship the project.

Do care about your people and make sure their morale stays high. A happy team is a productive team! Keep an eye out for small (or big) frustrations you can fix, or ways you can encourage/reward people for good work.

Are positive but realistic.

This is really hard to do when things aren't going well. Find little things to be positive about if the big ones aren't there.

But don't let your positivity erode your credibility: if things are going badly, people need to know or they can't fix it.

If things are going well, point it out and congratulate your team! If you have good credibility people will believe you and be happier.

Take care of business

Get out on the floor; moving information around requires that.

A good producer is like Batman: if they're doing their job well, you don't really notice them.

Good Habits and Bad Habits

Bad producers:

- Have an agenda
- Assume they know things
- Lack humility
- Use jargon when normal words will do
- Act like art or design directors
- Only value what they understand

Make the gameplay more emergent and procedural

Have an agenda

> They put something else before shipping the project: for example, they have a strong opinion about the art style and try to steer the style in the way they think it should go. Your job is not to determine the best art style for the project.

> Some producers may have additional roles that make them prioritize the company or people higher than the project, and this can be difficult to manage.

Assume they know things

> Are afraid of looking stupid

Lack humility

> Humility is huge. Be humble! Don't compete with your devs.

Use jargon when normal words will do

> Or other things to make themselves look/feel more important.

> Jargon is often used defensively, to distract attention from problems. Don't do this.

Behave like art or design directors

> Don't do that.
>
> When people are starting to act like art or design directors, sometimes that indicates a problem: that art/design direction isn't getting to the team.
>
> If you are an amazing art director, get a job as an art director. But don't use your job as a producer to do clandestine art direction.

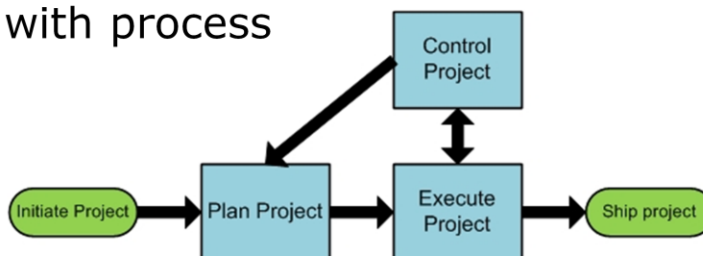Think that things they don't know about or aren't familiar with aren't important.

> I heard a story about a new executive who wanted to fire all of the 'scripting guys' because they didn't sound very important. The dev team said, "No, those are the people who actually make the game work, please don't fire them...!" Make sure you have all of the information before you decide whether something is important or not.

Remember that these things are almost never done maliciously. If a producer is telling people what they want to hear, or discipline directing, it's probably that they're trying to make things happen faster and have just chosen a harmful way to do it out of desperation. You won't wake up one morning and think, "I think I'll lie to people at work today," but you might think, "I'll tell this idiot what they want to hear so they get off my back and let me do my job." But having noble reasons for it doesn't erase the harm it does.

3. Fix Your Weaknesses

What were you missing in your former role?

- Knowledge of other areas of game development
- Greater responsibility (and authority)
- Familiarity with process

What were you missing in your former role?

Generally the biggest answer to this is: communicating with other departments. Particularly producers coming from publishing QA often have very little experience talking to devs at all (especially telling them what to do).

Knowledge of the other areas of the project

You will probably be more familiar with either 'upstream' or 'downstream' work. If you're an artist or programmer, you've probably worked farther 'upstream', i.e. you don't have a lot of dependencies but lots of people are dependent on you. If you've been 'downstream' in design, QA, tech art, or have been a support engineer, you're probably more familiar with projects as a whole and understand dependencies a bit better.

Make sure you get familiar with the other end of the pipe than the one you're used to.

Greater responsibility (and authority)

    If you're currently in another discipline, you probably

depend on your producer for all kinds of things that you don't even notice. When you go into production, you will need to do all of those things.

Be bold! If something needs to get fixed, you need to make sure it gets fixed. Probably nobody else will have time to follow up and make sure whatever it is gets done: this is your job now.

If something needs attention, it's your job to make sure it gets it.

I was fortunate to have been a lead before moving to production, so I was used to this, but if you haven't had a role with responsibility for others' work, be prepared.

Familiarity with process

That is, how you plan and control how the game gets made

Heavy processes, like big detailed project schedules or timesheets that everyone fills out each week, or agile processes, like scrum or taskboards.

These each have their place; there is no one-size-fits-all process. You'll gravitate towards what you're used to, or what worked for you last time, but don't force the wrong process onto a team just because you like it.

Particularly since scheduling styles change from one end to the other: 'upstream' tends to be more waterfall, and is much more about predictability: we'll make character A, which will take 4 weeks, then we'll do character B, which will take 4 weeks, etc. 'Downstream' needs to be more agile so the work can respond to changes upstream.

A quick word on process:

This could be a whole talk in itself. And there are other talks at GDC that cover it in more detail.

What determines how you plan and control work on the game?

What area of the game are you producing?

Is your team large or small?

Inexperienced or mature?

Remember that teams will tend to think they're more mature than they are, one of your jobs is to know if they actually are or not.

Start of project or end of project

Low rate of change (a sequel) or high rate of change

Regardless of the type of process you use, make sure that the team understands why it's necessary and that they trust it. Make sure the team gets a benefit from it. Remember that starting a process, and even getting the team to follow it, is the easy part: maintaining it and keeping it useful and relevant is the hard part.

How to Avoid Common Pitfalls

Telling people what they want to hear

- Stick to the facts, be honest
- If you don't know, find out
- This might be the hardest part of your job.

Telling people what they want to hear.

> Stick to the facts, be honest
>
> If you don't know, find out
>
> > Tell them you'll find out and get back to them later
> >
> > And then get back to them quickly! This is awesome for your credibility.
> >
> > My husband remembers vividly me coming home from work on my first production gig and being super excited that I did this: told someone I'd find something out and got back to them a few hours later and impressed them with my reliability. I was really proud of this, it built my credibility, and it's an easy win!
>
> This is the hardest part of the job for many people.
>
> It is hard to avoid doing this, especially if you have a facilitation/service mindset that is really beneficial to have as a producer.
>
> Doing this will not only destroy your credibility, it may also destroy the project and even the company. I have

seen this happen; it was tragic. Everyone was doing what they thought was right, but it had huge, negative ripple effects.
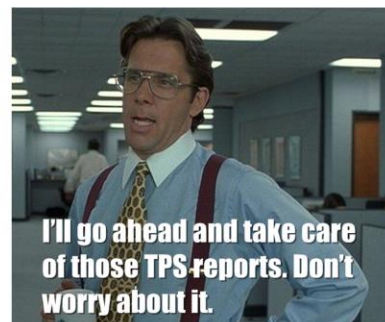
Here's why:

Often this happens because of wishful thinking, or being overly optimistic. Making a schedule that will get the game done on time IF everything goes well, nobody quits, there are no snow days, nobody gets pneumonia, etc. This never happens: schedules like that are lies. They tell people what they want to hear (Yeah, we can totally make this game in the time allotted!) but are not honest.

How to Avoid Common Pitfalls

Wasting the team's time

- Do everything you do with intention
- Make sure your meetings have goals
- Make things easy for people
  - Or do it for them

I'll go ahead and take care of those TPS reports. Don't worry about it.

Do everything you do with intention.

If you go to a meeting, know why you're there. To take notes and disseminate them? To provide information on tasking? To make sure a decision is made? To make sure you know the details of a decision so you can communicate it to your team?

Know why you scheduled that meeting

If you schedule a meeting, know for sure what its purpose is and what the outcome should be, and communicate that to all of the attendees. Meetings, for the most part, should be about making decisions. Disseminating information can usually be done more efficiently other ways.

If your daily standup is longer than 15 minutes, fix it. Do not make people stand and listen to 40 other people say 'no impediments' for an hour every day. Yes, this happens. Don't let it happen to your team.

Regular team meetings are an exception to the 'meetings should be about making decisions' rule

20

and help the project stay healthy. So are meetings about sensitive information that can't be e-mailed.

Make things easy for people

Or do it for them

Don't expect people to do things that are a pain unless you stand behind them and make them do it. If it's easy and has value, they'll do it. If it's hard and doesn't have any visible value, they'll avoid it if they possibly can.

How to Avoid Common Pitfalls

Blaming people for things they can't control

- Make sure people are empowered to fix problems
- This includes you

If someone is responsible for something, they need to have authority over it.

> This includes you. Make sure you have the authority to fix problems in areas you're responsible for.
>
> > And really, if you're responsible for an area, any problems are *your* problem. A problem might be out of your control but you can always do something to mitigate it, even if it's just to communicate the problem to everyone involved.
>
> Make sure that if someone on your team is responsible for something, they have authority/ownership over it.
>
> > They'll do better work if they have ownership, and being responsible for something you can't control is extremely frustrating.

How to Avoid Common Pitfalls

Poor team communication

- Recognize communication challenges
- Don't over-communicate
- Make your information easy to absorb

What Nobody Told Me About Production

- You know the cost of cool features
- Ideas are nothing without execution
- You are no longer an expert in your field
- You won't understand every detail
  - (and you don't really need to)
- Politics are part of production
- When things are going well, your job is easy

You know the cost of cool features

Prioritizing and measuring and disseminating information means you know the cost of cool features

> Everyone else on the team is thinking about cool features. You will have to be the first one to stop brainstorming and start pointing out how much things cost.

> "It would be cool if..." will put your teeth on edge, eventually

> "It would be so cool if all the walls in the game would explode when you shoot them and leave passageways!

>> "It would be cool if" is where a lot of cool stuff comes from, and you often have to oppose it.

>>> Changing it to "it would be incredibly cool if…" doesn't help.

Ideas are nothing without execution.

> Most of you are already very familiar with this. But as a producer you will be directly exposed to people who are farther removed from the project, and who are more

likely to have ideas without fully understanding how much those ideas cost in execution time (or how well you can actually execute those ideas). And they will need to get that understanding from you so they can ask for things that are actually doable.

Executing the project well will require the team to know what the values of the company are, and align their project priorities to those values. One of your responsibilities is to make sure that the ideas the team is having align to those values.

> Maybe your company wants to have the best graphics tech in the world, and that's their highest priority. That affects your team's work: if one of your world artists has some extra time, they could spend it doing R&D on the latest shader tech. When people are working on stuff that's aligned to the company's priorities, it means their work will be more valued (and less likely to get cut). You can give them that information, on values they can align to.

You are no longer an expert in your field

> And even if you are, people will assume you're not.

> Your old discipline will still keep going without you. You won't be able to rely on your previous knowledge as much as you might expect to.

> I always say that I'll only play the "I used to be a level designer" card once a year; so far I've stuck to that. Few things are more irritating that someone who assumes they know how to do someone else's job better than the person doing the job. This is the fastest way to make everyone hate you.

When people are talking about details, you probably won't understand what they're talking about. And you don't need to.

> You don't need to be an expert on everything (and games are too complex for this to be possible anyway).

> You only need to understand enough to be able to do your job, to know how long their tasks take and how easy or hard things are for them to do, and know what they need from you.

> It is totally ok (and actually most people will appreciate it) if you say, "I don't understand most of that, but it sounds like X

is more important than Y right now, and you need me to follow up with engineering on when Z, which is blocking X right now, will be done."

Or, it's ok to (occasionally) ask them to write up an e-mail outlining what they just said so you get the terminology correct when communicating it to others.

Do try to learn as much as possible (ask questions!) but be cool with the fact that you're not an expert.

Politics will be part of your job.

Politics is the art of influencing groups of people.

Your job is to ship the game. That involves getting a lot of people to do things they might not want to do, which is easier when you're good at influencing them. But at the same time, if you're on the game's side, and you have accurate information, you can use that to get other people to do the things they need to do to move the project forward.

I overheard an artist once say that they had liked me before I became "too corporate" which made me sad. Corporate? Me? But it's all a matter of degrees. The more you work with corporate political types, the more you become like them.

Being a producer when things are going well is easy.

Just stand back, grease the rails, and watch your team succeed! But: margins in this industry are small. Game projects rarely go **that** well. So if your job is easy, you are either lucky to be on a well-planned project with a great team, you're lazy, or you just haven't figured out what's going wrong yet.

In Summary

- Production is different
- Your background will give you strengths
- Don't let your mistakes become habits
- Be useful, be humble, and ask questions

Production is different

> Production is really different from other disciplines, and that may take some getting used to.

Your background will give you strengths

> identify them and build on them while filling in your weaknesses.

> Make your weaknesses into strengths by asking a lot of questions, and avoid making your strengths into weaknesses by assuming your background makes you an expert.

You will make mistakes

> Just don't let them become habits.

> A good manager or mentor will be a big help with this.

Be useful, be humble, and ask a lot of questions.

Q&A

Resources

- GDC Talks:
  - From Great Ideas to Game Features
  - Kanban - How to Make Your Production Scream
  - The Planning Fallacy
  - Debunking Myths About Lean Game Development