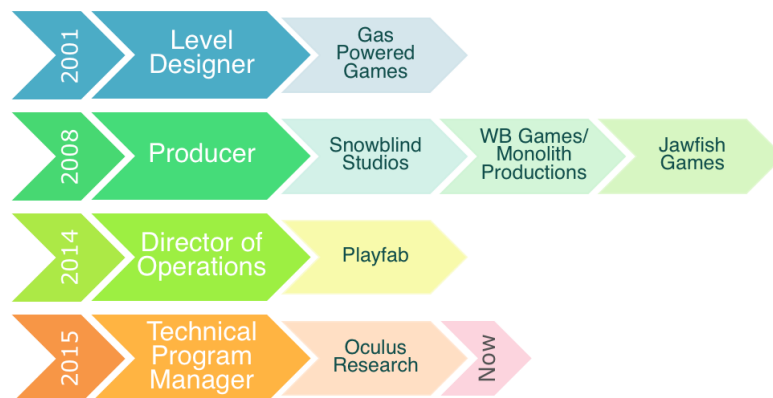




Hello, and welcome to Producer Bootcamp!

Silence phones, fill out surveys, email me instead of taking pictures of the slides. I'll leave lots of time for Q&A at the end, so please keep track of anything you want to know more about!



Hi, I'm Ruth Tomandl and I've been a producer/PM for 10 years (and 3 days!), on 8 different teams at 5 companies.

I started in the games industry at Gas Powered Games as a Level Designer working on the Dungeon Siege games, and I really liked scheduling and coordination so I looked for a Producer job and joined a small independent game company, Snowblind Studios, which was acquired by Warner Brothers soon after and merged into Monolith Productions. In 2013 I joined a mobile game startup, and in 2014 it shut down and I helped Uber Entertainment start a backend startup called Playfab.

I'm currently at Oculus Research, and we are growing really quickly. Lately I've been doing a lot of interviewing, hiring, managing, and mentoring, and figuring out which skills are needed on which teams.



Overview

- The job of a producer
- Different producer strengths
- Different types of teams
- Debugging your skills and building your strengths



In this talk, I want to share some things I've learned about strengths that I wish I had known 10 years ago: what I've learned about producer strengths in general, my own strengths and how they've fitted the teams I've been on (for better or for worse), and how our strengths and weaknesses affect what kinds of producers we are.

So today, I'm going to talk about:

- The job of a producer: what even is it?
- Different types of producer strengths
- Different types of teams (and what kinds of producer skills are needed on which kinds of teams)
- How to debug your own skills to better fit the needs of your team and build your strengths.

What Does a Producer Do?

First: To understand which producer strengths are important, we need to talk about what a producer actually does.

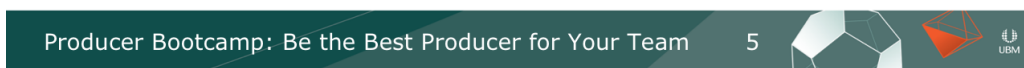
How many of you are currently producers on a game team? How many are game developers but not producers? Students? People who aren't producers yet, but want to be?



The person who makes sure the game gets **done**.



Producer | Project Manager | Program Manager
Product Manager | Business Analyst | PM
Scrum Master | Operations Manager



The job of a Producer is to make sure the game gets done. “Done” is a very important word; I've bolded it and put it in red and I'd put little fireworks around it if I could. Games that don't get done don't make money, and if you don't finish games, you don't usually get to keep making games.

Obviously that job covers a lot of possible ground. There are also a lot of different names for that type of job: depending on your team or company and how they define roles, you may be called: Producer, Project Manager, Program Manger, Product Manager, Business Analyst, Scrum Master, Operations Manager

If you look each of these up, the exact definitions will all be slightly different, but they all basically fall under the umbrella of ‘people who make sure the game gets done’. And sometimes the definitions for these jobs is different at different companies, so for this talk I'm just going to say “producer”.

So what does this job entail?



Making Sure the Game Gets Done

- Prioritize
- Measure Progress
- Drive (decisions and tasks)
- Communicate

SHIP IT

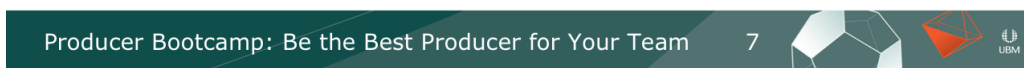


There are a few main things a producer does. There's a ton more, of course, and exactly what you do will depend on your skills and your team's needs, which we'll talk more about, but these are all things that producers need to do.

- First, Producers prioritize. They keep an eye on and communicate what work is the most important to getting the game done. You can't do everything because you always have some constraints, so it's important to make sure the important things get done so your game gets done.
- Producers measure progress, they measure work getting done. It's very hard to finish something if you can't measure your progress towards your goals. Especially if your constraints (like budget or schedule) are very tight, you need to be able to accurately measure your progress so you can be confident you'll be able to finish the game within those constraints, or to make changes to your plan if you have to. You want to see problems coming as early as possible, and have a good idea of what you'll really be able to accomplish.
- Drive decisions and tasks: making sure work gets done and decisions get made when they need to
- Communicate: they get the right information to the right people so decision-makers can make the right decisions and the team can do the right work.



- Stay calm and anticipate problems
- Ask lots of questions



These are some skills that I think every good producer needs to have to do that work well.

Remain calm, and be prepared when a crisis happens. Take things in stride. If you react badly, people will stop coming to you with problems or bad news, which will keep you from being able to do your job well since so much of your job is collecting and distributing information.

- I once 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. Save up your passion and emotions for when good things happen!

Ask lots of questions and listen a lot: especially, be willing to ask questions that don't make you look good. One useful thing about being the only non-engineer in a meeting is that you can be the person who's willing to ask 'stupid' questions that engineers or designers might be afraid to ask. Very often, when I do this, someone on my team will come up to me afterwards and say "thank you, I didn't know the answer to that question either but I was afraid to ask and look dumb in that meeting". You can have a lot of positive impact on your team this way!



Skills All Producers Need

- Stay calm and anticipate problems
- Ask lots of questions
- Care about your project and your team
- Give bad news and speak the truth



Care about your project and your team. The game always comes first; it's important to care about your team, and a happy team does better work, but I've seen producers who protect their team too much at the expense of the project, or who don't want to give critical feedback or talk about problems because they don't want to hurt someone's feelings. If the game doesn't get done, you don't get paid, which will hurt everyone's feelings much more in the long run.

Bad news is usually the information that's most critical to your team, especially to people making decisions. Often you'll be the one who finds out about problems, and you need to be comfortable sharing them. And if you never give bad news or negative feedback, people will be less likely to believe you when you say positive things: people who only say how great everything is all the time lose credibility really fast. Make sure the bad news you're sharing is constructive, that is something that can be acted on, and not just whining or complaining, but this will be a critical part of your job.

Skills All Producers Need

- Stay calm and anticipate problems
- Ask lots of questions
- Care about your project and your team
- Give bad news and speak the truth
- Get things done¹
- Own and communicate your game vision

1. Alison Green, "do what you say you're going to do", *Ask a Manager*, 3/10/2011

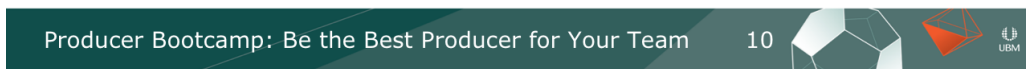
Get things done. Do the things that you say you'll do, when you said you'd do them. This is more uncommon than it should be. Don't just start things: follow them all the way through until they're finished. [1: <http://www.askamanager.org/2011/03/do-what-you-say-youre-going-to-do.html>]

Own and communicate your game vision: you should be able to explain to each person on your team how the work they're doing fits into the overall game vision, why decisions that affect them were made, and what they should expect in the near future.

These are skills that every *good* producer should have, but:



What do you think is the difference between
a **good** producer and a **great** producer?



I'd like to ask everybody to think about what in your opinion is the difference between a decent producer and a great producer. This is a question I ask everyone I interview, and I think it's one of the most important questions we can ask ourselves. Think about your answer, and write it down or remember it for later.

I'll give you a second to think about this and write it down.

Producer Strengths

I think the answer lies in the difference between skills and strengths. So what do I mean by that? What's a strength?



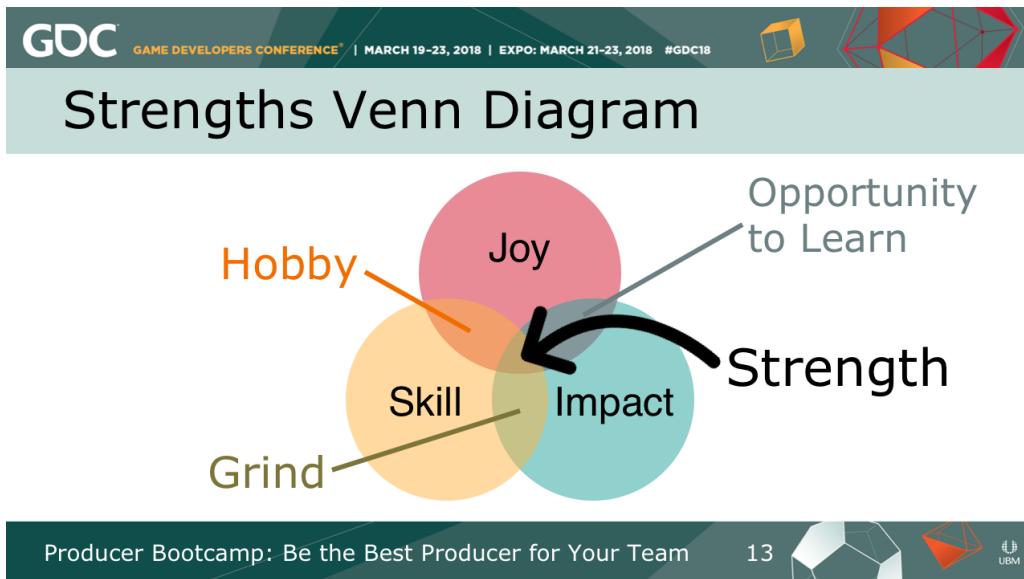
Strengths vs. Weaknesses

Strengths are things you are good at, that you love doing, **and** that are useful in your current job

Weakness are things that you are bad at, you hate doing, **or** are not useful in your current job.



If that's not entirely clear, don't worry: I have a Venn Diagram.



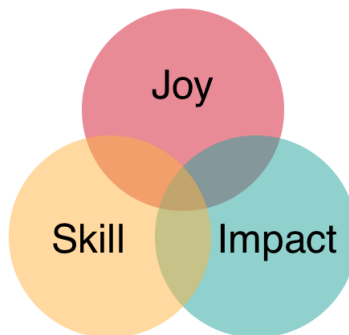
Here's a diagram to make it clearer. If you love doing something, and you're good at it (i.e. you have skill), and it has positive impact on your game, then it's a strength.

[Note that these are very similar to the Vitruvian Triad:

<http://art3idea.psu.edu/locus/vitruvius2.pdf> or the components of ikegai:

<http://upliftconnect.com/ikigai-finding-your-reason-for-being/>]

1. If you love it and you're good at it, but there's no need for it in your current job, then it's a hobby.
2. If you love it and there's a need for it but you're bad at it, that's an opportunity to learn (or help out on easy stuff). But it's not a strength, at least not yet, but it's an opportunity to develop a strength.
3. If you are good at something and it's needed to finish the game, but you hate doing it, then it's not a strength. It can be sustainable to do that work, but most people won't be happy staying in a job where it's the majority of what they do. Face it: game development doesn't usually pay well enough to keep you in a job that doesn't bring you joy. I'm not saying you have to be in a constant state of euphoria, but your job should bring you fulfillment and satisfaction overall, and you shouldn't dread going to work.



One good habit is, every 6 months or so sit down and put numbers to these things. What percentage of your time do you spend doing work you enjoy? What percentage do you spend doing things you're great at? What percent do you spend doing things that have a big impact on your game project or studio? If any of those numbers are low, figure out why. Is it because you've changed roles recently and are still struggling to find your feet and get good? Is it because things are just hard right now but there's a light at the end of the tunnel and things are getting better? Is it because your team really needed someone to manage localization and you volunteered because it needs to be done but it's the least enjoyable job you've ever had and you can't wait to get out of it?

Find a path to raising those percentages, and set clear goals to improve them. If you have to do something you hate, find the joy in it if you can. Maybe you hate localization, but you love knowing that you're making the loc team's jobs easier by making sure the text is being delivered in the right format.

If you can find the joy in your work, you'll do a better job, be more valuable to your team, and be able to stick with it long enough to get great at it.



Status Owner



Facilitator



Inspirer

Remember the question I asked, about what makes a producer great? These are three examples of producer specializations, or subclasses, and generally I've seen answers to that question fall under one of these. For example, I've been told that a great producer is someone who is great at communicating the game vision to their team and inspiring them to believe in it.

You might be strongest in one of these subclasses, or you might be multi-specced. Let's talk about what each of these types of producers do, and where they shine.



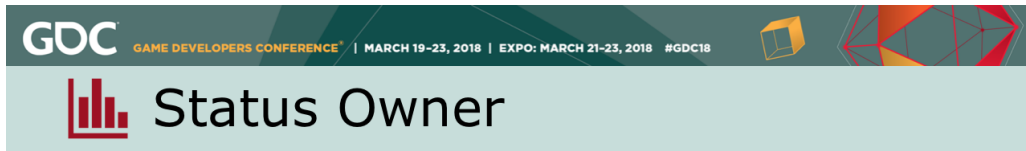
Status Owner

- Knows everything there is to know about the status of work being done on the game
- People come to them for information
- They make sure communication is happening
- Shares data in ways everyone can understand
- Makes docs people actually want to read!



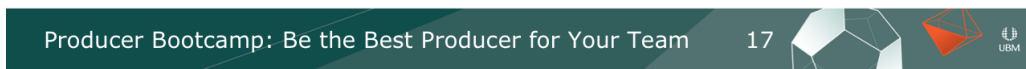
The Status Owner is the person who knows everything there is to know about the status of the game. Where are the design docs, and are they up to date? How long does it take us to build a medium-sized level? How many characters can we build between now and ship? Are we on track to hit the next 3 milestones?

If you're nosy and love Microsoft Project, and love sending very clear, color-coded emails outlining progress towards the next milestone, or if someone's ever said to you, "wow, that schedule is beautiful" you're probably a Status Owner.



**You can have data without information, but
you cannot have information without data.**

-- Daniel Keys Moran

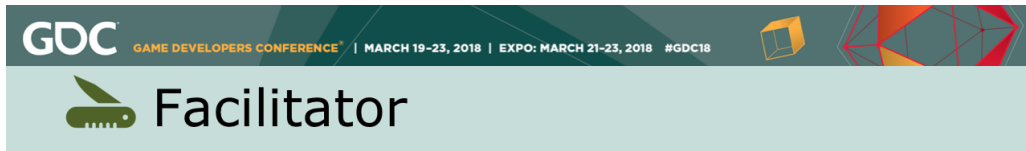


Or if this quote resonates with you. 😊

- Makes sure their team has what they need to succeed
- Jumps in and gets hands dirty: no task is too small!
- Unblocks team, is always fixing blocking issues
- Makes everything 'just work'
- Doesn't just solve problems: anticipates them.

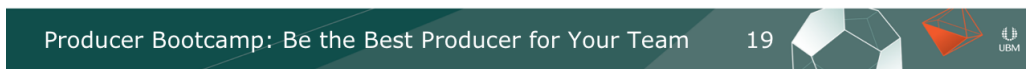
The Facilitator is the person that people go to when they need help, when they're blocked or stuck. They are reliable and have the trust of the team.

If you hear the QA team is running out of desk space and your first thought is how quickly can you make it to Ikea, you're probably a facilitator.



The most meaningful way to succeed is to help others succeed.

-- Adam Grant



If helping others succeed is what really brings you joy, you're probably a facilitator.

"But wait," you're saying. "What if I love Microsoft Project *and* I love building Ikea desks?" Remember, these are not mutually exclusive skill sets. The more of them you have, the more situations you'll encounter where you'll be a great producer. But nobody's good at everything. Speaking of which, now that we've covered two areas I'm personally really good at, let's talk about one I'm weak in.

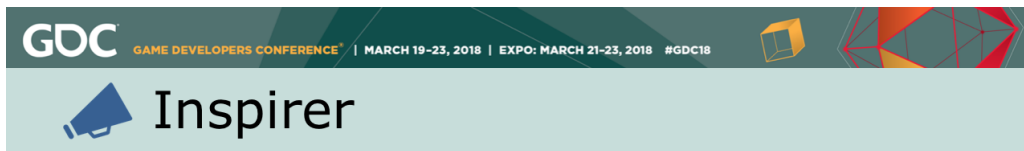


Inspirer

- Owns the narrative and project vision, helps the team understand why work is being done
- Cheerleader for the team
- Helps the people on their team learn and grow
- Is a leader and an agent of change
- Listens to their team and is a 'safe place' to talk

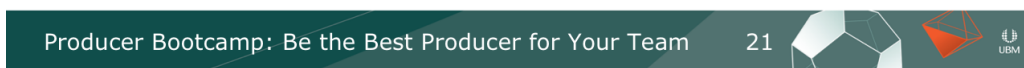


Do you love telling people about your game? Do you get excited about what your team is doing and think they're capable of doing even more? Do you often brag about a cool thing your team just did? When your team members come to ask you about the game, have an epiphany while talking to you, and walk away reinvigorated and more excited, you're an inspirer.



If you want to build a ship, don't drum up people to collect wood and don't assign them tasks and work, but rather teach them to long for the endless immensity of the sea.

-- Antoine de Saint-Exupery



This is what inspirers do: they help their team yearn to build a great game, and they help their team envision what that game will be, and to believe in that vision.

If your team is doing work-for-hire to make ends meet, and you're working on porting a teddy bear rhythm game to iPad, this person is somehow able to get you excited about making the best darn teddy bear rhythm game iPad port the world has ever seen.

I've had the joy of working with a lot of these people over the years, but sadly I am not one of them. I tend to be far too pragmatic and data-driven: yes, this team is capable of executing an adequate and profitable teddy bear rhythm game port; here's a first-pass schedule. That approach gets things done, but it's not exactly inspiring.

I love working with inspirers, and I especially love it when they appreciate what I do and when we work together well. Especially on big teams, you'll have many types of producers. Having a good understanding of your own strengths can help you understand the strengths of the other producers on your team and see how best you can work together.

So what are your producer strengths? How do you identify and grow them?



What are Your Producer Strengths?

1. What do you think makes a producer great?
2. What brings you joy?
3. What is your superpower? What does your team praise you for?



First, think back to your answer to the question about the difference between a good producer and a great one. Was your answer something that you already do that makes you great? Is it something you wish you were better at, or a strength you see in a coworker you admire? Is it something that your team needs right now? Whatever your answer is, it is probably a clue: it might tell you what producer strength you value the most, which can give you an idea of what strengths you should focus on building in yourself.

Second, ask yourself: what about your job brings you joy? What do you look forward to being able to do every day? For me, the best thing in my day is when someone comes to me with a problem that's blocking them, that I'm able to solve quickly for them to get them unblocked.

Third: Identify your producer superpower. What would someone on your team say is the reason they're glad you're on the team? What does your team praise you and thank you for?



What are Your Producer Strengths?

1. What do you think makes a producer great?
2. What brings you joy?
3. What is your superpower? What does your team praise you for?
4. What does 'success' look like for you?



And think about what 'success' means for you in your role. That is, what is used to measure whether you're doing well? Becoming the best possible producer for your team requires understanding what skills you're expected to build. Is "success" for you that your team has clear milestones and knows what they need to do to hit them? Is it that the leaders on your team know what everyone's doing well enough that they can confidently make the right decisions? Is it that your team is finishing 3 levels per month?

The answers to these questions should tell you what strengths you already have, and which strengths you want to improve.



Status Owner

- Take a PMP or Scrum class (yes, really)
- Build and share a strong doc template collection



Here are some suggestions on how to build strengths for the three types of producers we just talked about; I'll talk about general strength building later in the talk, as well.

For the Status Owner: Take traditional project management classes. Even the ones that game developers often make fun of, or think are useless. The Gantt chart is over 100 years old, and project management has been around since humans started building things in teams. There's way more useful knowledge in existence than you can figure out on your own while also trying to ship a successful game.

And level up your document template game: whenever I hear someone say, "Wow, that's a really great schedule!" the next thing they say is, "Can I get a copy?" Over time you can build up a portfolio of useful status document templates that you can have ready to share or use.



Status Owner

- Take a PMP or Scrum class (yes, really)
- Build and share a strong doc template collection



Facilitator

- Learn side skills (scripting, Unity, VO direction)
- Run post-mortems to get better at finding problems

Facilitator: Filling holes is easier when you know how to do more things. Especially things nobody wants to do, like spending days at a remote studio directing actors recording VO. Make sure it's a skill that's actually useful for the project, though: I once started learning Elvish when I was working on a Lord of the Rings game, until the writer told me it was really not necessary and to please stop unless I had a deep desire to know Elvish.

Another strength Facilitators can develop over the long run is running post-mortems. This is an example of one of those things that everyone thinks would be valuable, but that nobody wants badly enough to figure out how to do well. Keep your ears open for the phrase "we should really do X..." and then figure out how to do whatever X is if nobody's jumping on it and your team needs it.



Status Owner

- Take a PMP or Scrum class (yes, really)
- Build and share a strong doc template collection



Facilitator

- Learn side skills (scripting, Unity, VO direction)
- Run post-mortems to get better at finding problems



Inspirer

- Be a mentor to junior Producers
- Try media, public speaking, or community management

Inspirers make great mentors, and mentoring will help you grow your own producer skills too. Coaching someone and helping them develop a strength is a great way to get better at that strength yourself.

Inspirers also make great media spokespeople, public speakers, and community managers. I did a lot of interviews and media for a couple of the games I worked on, because nobody else wanted to do it so I volunteered. I wasn't amazing at it, but after doing it for a while I was a thousand times better than I was when I started, and it's a useful thing to know how to do, and it helped my team. If you're already inspiring your team to believe in the game you're building, why not inspire players to believe in it, too?

GDC GAME DEVELOPERS CONFERENCE | MARCH 19-23, 2018 | EXPO: MARCH 21-23, 2018 #GDC18

When Strengths are Weaknesses



Nega-Inspirer

- Prioritizes quality until it endangers the schedule
- Is artificially cheerful, tone-deaf

Producer Bootcamp: Be the Best Producer for Your Team 27



When building skills, always remember to pay attention to whether they're the ones your team needs. Sometimes a skill that's a strength on one team can be a weakness on another. A producer can overuse a strength of theirs to the point where it starts having a negative impact instead of a positive one.


I think we've all been on a team with someone who loves doing things a certain way, and maybe is very good at it, but it's just not what the project needs right then.

Some examples that I've seen:

Inspirers who are so focused on encouraging the team to make the highest possible quality game that they endanger the schedule. Especially when the schedule is tight and inflexible, or the risks are high, this misprioritization can be really dangerous. You might be at risk of doing this if you're internally yelling at me right now, "What am I supposed to do, then: ship a bad game?!" No, but remember: if your game doesn't ship, it doesn't matter how perfect your volumetric lighting is.

Inspirers can also fall into the trap of thinking that their job is to be cheerful and positive no matter what. If the project is clearly going off the rails and your team is legitimately worried about real problems, you being a happy ray of sunshine every day may make them stop trusting what you say, if you don't ever talk about the problems. And if your team doesn't trust you, you can't do your job.

GDC GAME DEVELOPERS CONFERENCE | MARCH 19-23, 2018 | EXPO: MARCH 21-23, 2018 #GDC18



When Strengths are Weaknesses

Nega-Inspirer

- Prioritizes quality until it endangers the schedule
- Is artificially cheerful, tone-deaf

Anti-Facilitator

- Fills holes that are too big (e.g. art directing)
- Causes disruptions where they don't belong

Producer Bootcamp: Be the Best Producer for Your Team 28



Facilitators are really good at jumping in and filling holes and getting things done, but sometimes they fill the wrong hole and bandaid over a problem that really needs to be fixed the right way. I was on a project that didn't have an art director, and since the producer had a bit of an art background, they thought they could jump in and help by art directing. The problem was that they were trying to do two really difficult jobs, and because they had authority as a producer it was hard for the team to question their art direction, and it really hurt the overall quality of the game (and the schedule, since the producer was willing to tolerate delays in order to get their own ideas into the game). I'm not saying 'stay in your lane', or 'don't help your team', but I am saying that if you are jumping in to fill a really big hole, step back and make sure that's really the best thing for the game and your team, and not just a fun opportunity for you to use some skills you don't usually get to use.

Similarly, sometimes you'll want to jump in and help in an area that doesn't really need it, or with something you're just really bad at. Always make sure you're actually being helpful and not just disruptive.

GDC GAME DEVELOPERS CONFERENCE | MARCH 19-23, 2018 | EXPO: MARCH 21-23, 2018 #GDC18

When Strengths are Weaknesses

**Nega-Inspirer**

- Prioritizes quality until it endangers the schedule
- Is artificially cheerful, tone-deaf

**Anti-Facilitator**

- Fills holes that are too big (e.g. art directing)
- Causes disruptions where they don't belong

**Dark Status Owner**

- Checks boxes but misses larger issues
- Focuses slavishly on deadlines and process

Producer Bootcamp: Be the Best Producer for Your Team 29

Sometimes Status Owners can accidentally miss larger issues because they're so focused on minutiae. Have you ever been on a game where every week, every milestone, everything was 'on schedule' and then suddenly you were way behind or some feature just didn't work? Marking tasks as 'done' is not the same thing as finishing the game, and telling people everything is on schedule when it's not is one of the worst mistakes you can make.

And of course, status owners who are slavishly devoted to schedules, deadlines, budgets, or processes over everything else. This is usually what people are complaining about when they complain about producers. I know I keep saying the game needs to get done, and it does! But most people in game development aren't there because they love hitting milestones. They're there because they love making great games, and if they see you as an obstacle to that, they'll hate working with you.

Teams

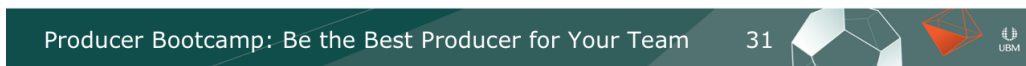
So which strengths are the right ones for your specific project? A lot of that comes down to what your team is like, and what they need from you.



Things All Teams Need

1. A viable, compelling, clear, and well-communicated shared vision for what you're making²

2. Paul Tozour, "The Game Outcomes Project", *The Intelligence Engine*, 12/19/2014



First of all, there are some things that all teams need in order to ship a successful game. Since your job as producer is to make sure the game gets done, it's your job to make sure your team has these things.

If you haven't already, I strongly encourage every producer to read the Game Outcomes Project [1: <http://intelligenceengine.blogspot.com/2014/12/the-game-outcomes-project-part-1-best.html>], which was published a few years ago. Paul Tozour and his team surveyed hundreds of game developers asking about their development process, teamwork, and culture, and comparing those responses to the success of the teams' finished games. The results are extremely interesting and there's a ton of useful information there, but their strongest result was that teams that had a "viable, compelling, clear, and well-communicated shared vision" were the most likely to make a successful game: critically successful, financially successful, and meeting or exceeding the team's own expectations for what they were making.

[Their complete list of 40 great team behaviors can be found on Gamasutra: https://www.gamasutra.com/blogs/PaulTozour/20150126/235024/The_Game_Outcomes_Project_Part_5_What_Great_Teams_Do.php]

Things All Teams Need

1. A viable, compelling, clear, and well-communicated shared vision for what you're making²
2. Management of risks to the vision and plan
3. Mutual respect so ideas can be shared freely³
4. Understanding how their work fits the vision, buy-in

2. Paul Tozour, "The Game Outcomes Project", *The Intelligence Engine*, 12/19/2014

3. Charles Duhigg, "What Google Learned From Its Quest to Build the Perfect Team", *The New York Times Magazine*, 2/25/2016

Second, protecting your game's vision, and your plan for executing on that vision, from risks. A lot of this falls on the producer to identify those risks and make backup plans.

Your team needs healthy ways to communicate and share ideas. Google's Project Aristotle is also a great resource for understanding how to foster an environment of mutual respect and psychological safety: 2:

<https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html>

Game development is very collaborative, and if your team members don't feel comfortable sharing their ideas, you'll miss out on what they'd be able to contribute on a team where they felt respected.

And people on your team need to understand what you're building and how their work fits into the overall vision. They need to believe in what you're doing and feel bought in.

Every team needs these things, but successful teams can take a lot of different forms and be structured a lot of different ways. Let's go over some different team types, and talk about how each of our three producer specs can help those teams be successful.



Three Types of Teams



Strict Chain of Command



Strike Team Collective



Soup of Experts



I'll talk about three types of teams: teams are very structured with a clear chain of command, where decisions are made at the top; teams built out of smaller teams that each have ownership over a specific area, and a 'soup of experts' team (if you have a better name for this, let me know) which is a flat or unstructured collection of very high performers who each have a lot of autonomy.

One thing you may notice is that the team structure usually has a lot to do with the size of the team. The larger a team is, *usually* the more structure it needs to be effective. And smaller teams tend to be flatter and less structured. Most 400-person AAA game teams will be a strict chain of command, and most 5-person indie teams will be a soup of experts.

Most teams will be a combination of different types. I worked on a game where the design team was a set of strike teams, the art team was a rigid dictatorship, and the engine programming team was a flat, unstructured soup.

Even though reality is always messier than a PowerPoint presentation, and your team probably won't exactly fit one of these descriptions, thinking about teams this way lets you recognize where you have experience and where you don't. If you've mostly worked on one type of team, you'll naturally adapt your skills to fit what that type of team needs from you.



Strict Chain of Command

- Decision making is centralized
- Team leads own their team's work
- **Pros:** scales well, communication channels are clear, structure is easy to understand
- **Cons:** difficult to change things and fix problems, team members may not feel heard



- Decision making is centralized, often one person or a few people are ultimately in charge of all decisions
- Team leads own their team's work and are responsible for it. They make decisions about their teams work but are accountable to the people at the top of the chain.
- Pros: scales well, communication channels are clear. The structure is easy to understand and it's clear to everybody exactly what their job is and how to do it well. It's very easy for junior people to join this team and thrive, because it's very clear to them what they need to do to be successful.
- Cons: difficult to change things or fix problems because this type of team has a lot of inertia and decisions can take a long time to make, team members at the bottom of the chain may not feel heard or like they have any say in what the game is



Status Owner

- Pass information up and down the chain
- Make sure decision-makers have the right data



Facilitator

- Help your team know where to find information
- Watch for holes (hint: the ones you try to fill)



Inspirer

- Make sure team understands the project vision
- Communicate how each piece of work fits in

So what can different kinds of producers do to make this type of team successful?

It's easy to do well on this type of team as a Status Owner, since accurate communication of data and numbers is what makes this type of team tick. The decision-makers at the top need to know what's happening at the bottom, so they can make the right decisions, and the people on the bottom need to know what decisions have been made at the top so they can do the right work. Producers are often the main people communicating this information up and down the chain.

A facilitator's strengths -- jumping in and helping or wearing a ton of different hats -- are frankly not as useful on this kind of team. When I work on this kind of team, I usually rely much more on my status owner strengths. Two things a facilitator can do to be effective on a strictly structured team are: 1. reduce the process load on your team by helping them know where things are and what they need to do (because these teams are usually heavier on process) and 2. watch for where you're trying to put on a new hat: if you're trying to take on something really outside of your role, that probably means there's a big hole that needs to be fixed. I mentioned earlier about producers art directing: occasionally it is the best thing for the team, but usually it's not, and on this type of team it almost never is.

Inspirers can really help the team understand how their work fits in; people do their best work when they understand why they're doing it and what they're working towards. In a strict chain of command, that 'why' can get lost on the way down the chain, and people can feel really separated from the game vision. I think this is the type of producer who really shines on this kind of team, and who can be the difference between a happy, productive, successful team and a team that stamps out something they're not particularly proud of.



Strike Team Collective

- Smaller decisions are owned by the teams doing the work, larger decisions might be top-down
- Teams are responsible for predictable delivery
- **Pros:** scales well, high ownership, works for various team experience or maturity
- **Cons:** communication and alignment between teams can be challenging



A project organized into strike teams or scrum teams leave most of the day-to-day decision making (and maybe even bigger decisions) up to a bunch of autonomous teams. Each team is responsible for its own work, and for delivering finished, working features and content to other teams or into the final product.

This structure has a lot of advantages and has gotten really popular over the last decade; it scales well, it allows small sub-teams to have ownership over their work and be more responsive.

However, it can be really hard to get information between teams, and to make sure everyone is working towards the same overall goals.



Status Owner

- Own inter-team communication
- Act as single point of contact for your team



Facilitator

- Make sure smaller teams have 'just enough' process
- Take (usually light) process burden off your team



Inspirer

- Empower and inspire your strike team
- Make sure teams are aligned to overall vision

A producer who is a strong status owner is vital for inter-team communication in a strike team collective. Autonomous teams are often great at trust and info sharing within the team, and really bad at sharing between teams. Especially if constraints are tight, having a single person who can speak for the team and knows everything about it is crucial. You can help build strong communications channels to other teams with your awesome status docs.

Facilitators on these kinds of teams often focus on finding just the right amount of process for their team, and then taking as much of the load of that process off of the team as possible. Agile development is often very light on process and facilitators can help make it even lighter.

Because strike teams have so much ownership of their own work, sometimes the vision isn't always aligned between the different teams. They might have different priorities or different ideas of what the game should be. An inspirer can help bang the drum of what the game vision is, escalate any misalignment they see, and get everyone marching in the same direction.



- Individuals own their own decisions and work
- Communication is peer-to-peer and ad hoc
- **Pros:** Process needs are minimal, very easy to change course
- **Cons:** Can be difficult to: measure progress and monitor status, keep on course, implement process



This is what you'll often see on very small, independent teams of veteran developers, or on very expert teams within a larger project. Individuals have a lot of ownership over their own work, and the structure is very ad-hoc. Each individual shares or seeks out information as they see fit.

This type of team has similar pros and cons to strike teams, but exaggerated. Very little process is needed, decisions can be made super quickly.

And measurement and communication can be difficult or impossible when each person knows about the status of their own work but doesn't necessarily write that information down. Alignment can be a huge challenge, especially if you're working with a lot of strong personalities with different opinions and no clear authority structure. It can be really hard to implement process when each person is empowered to tell you to go away.



Status Owner

- Make the implicit explicit: write things down
- Get creative with minimal and unusual process



Facilitator

- Get team what they need, get out of the way
- Fill in holes so experts can keep doing expert stuff



Inspirer

- Make sure everyone feels heard and supported
- Align (strong, independent, autonomous) individuals

On this kind of team, status owners can really help make the implicit explicit. That is, these teams often have strong connections between the contributors, and they don't think they need to write things down or have process because they just all understand what everyone else is doing and it feels really natural. But: because these teams haven't spent time building processes, it can be easy to help them add a few minimal things here and there, like a task tracker or project goal documentation, that they'll really appreciate. Find out what their pain points are, and add processes that help fix them. And often people outside of the team often has no idea what's going on inside the team beyond 'seems to be working', and if they need to know, you can be the one to communicate that information.

Facilitators are usually the superstars on this kind of team. Because each individual team member has so much autonomy and ownership, it's extra valuable to help them spend the maximum time doing what they're great at. When things are working well, a facilitator is great at making them better, and when they're not working well, a facilitator is great at tracking down the reasons why.

These teams are usually pretty great at inspiring themselves, but because they're so unstructured it can be easy for people to feel like they're being overlooked or unheard. An Inspirer can keep their finger on the emotional state of this team and make sure that there aren't major problems hiding beneath the expertise and autonomy. For example, individuals on these types of teams can feel a lot of pressure to act like they know what they're doing even if they feel uncertain about their ability to handle it, and they may not feel comfortable speaking up about that if they think everyone around them is awesome and confident.



Interviewing a Team

1. How is your team structured?
2. How are decisions made? Who makes them?
3. Who is the best producer on your team now, and why are they the best?



Most of us will change jobs at some point, and some of you are students on the hunt for your first producer gig, so: how can you figure out whether the team you're talking to is one you'd be effective on, or one that needs the types of strengths you have or want to develop? Here are some questions you can ask to help determine if the team is a good fit for your skills.

How is their team structured? Can they draw a diagram for you? I draw diagrams of my teams constantly; it's a really useful communication tool generally and it's great for explaining to candidates what they'd be getting themselves into.

How are decisions made? Who makes them? Are decisions made by a single Design Director, or by the people doing the work? What types of decisions are centralized, and what types are owned by individuals?

Who is the best producer on their team now, and why is that producer the best? What strengths does that person have that are so valuable to their team? Are those strengths that you have or want to build?



Interviewing a Team

1. How is your team structured?
2. How are decisions made? Who makes them?
3. Who is the best producer on your team now, and why are they the best?
4. What do you need most from a new producer?
5. What's the hardest problem your team is facing?



What do they need most from a new producer? Do they need you to jump and track progress towards a milestone? Do they need you to align two teams that don't agree on the game vision? Do they need someone to be generally helpful? What strengths are they looking for in candidates?

What's the hardest problem their team is facing? This will tell you where you can have the most impact on that team.

Make sure when interviewing that the type of producer they're looking for is a type that you want to be. I've discovered that I don't find joy in 'selling' how great my team is, or 'believing my own hype', and that's a strength that you need to have to be effective in many small, scrappy startups that need to promote themselves aggressively. If the sense I get when interviewing is that that's the type of producer they need, I'm not going to be as interested in that job.

Debugging

Ok, so we've talked about types of strengths and types of teams. So how do you know if your own strengths are a good match for your team? How do you figure out: what your team needs, what strengths you should build, and how do you build them?



How to Build Strengths Now

- Some strengths are always needed and some are situational: focus on the ones that are useful now
- Default to 'Yes' for trying new things



The easiest strengths to build are ones your team needs right now. Look for things that you can learn that will be widely useful, like building long-term schedules, but also things that are more situational but that your team really needs, like learning how to run an external localization team.

Second, default to 'yes' for trying new things. If your team is like most teams, there is so much work that needs to be done that you'll have a lot of opportunities to try things you've never done before. Remember that a strength is something you enjoy and are good at, but how do you know if you like something or are good at if you haven't tried it? A few years ago, I thought I wouldn't like or be good at being a manager, but because my team needed more managers, I gave it a try. Now I'm managing 8 people and I love it; I've learned a ton and it's a way I can have a lot of positive impact. Over my career, I've tried to always say 'yes' to new opportunities, and I've never regretted it. Even the times when I have really hated whatever the new thing was, now I know what types of jobs I hate, so I can avoid them and spend more time doing the things I love!



How to Build Strengths Now

- Some strengths are always needed and some are situational: focus on the ones that are useful now
- Default to 'Yes' for trying new things
- Work on as many different types of teams and projects as you can, especially early in your career

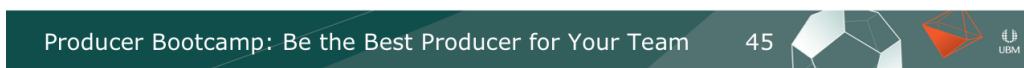


Speaking of new opportunities: You should work on as many different types of teams and projects as possible, especially early in your career. Moving to a new type of team that you don't have experience with is always a learning curve, and the more you do it, the more strengths you'll be able to build and the better you'll get at building new strengths. Everyone I've worked with who's only worked on one type of team has big blind spots, and sometimes they're hard to identify and fill in. For example, I worked with a producer who had only worked at a big software company; he had been there for decades. When he joined our very small, agile team, he didn't know how to jump in and be useful, because that had never been part of his job before. The longer you spend at one company, especially early in your career, the more your strengths will adapt to that company's specific needs and you might stunt your strengths' long-term growth.

I'm not telling you that you should quit your job! But recognize that you might be missing something if your experience isn't varied enough. There are probably strengths that you don't have, that you'll need later in your career. But there are other ways to build these strengths besides changing jobs.



- Take classes
 - *Even traditional project management classes*
- Find a mentor
- Talk to other types of producers on other types of teams
 - *Hint: You're at GDC Producer Bootcamp literally right this second.*



Take classes, even really boring project management classes. Often game developers look down their noses at Project Management Professional certification or Scrum Master certification, but as a new producer I took a project management class with PMs from Expedia and Amazon and realized that they faced the same kinds of problems I did, but in a slightly different context; it was really was eye-opening for me. Game development is special, but it's not *that* different from other software development when you get down to the nuts and bolts and problems people face.

Find a mentor with different or more varied experience than you, or who's good at something you want to get better at. I know that you'll hear this a lot: "Find a mentor!" and you don't really know how to do it. So: how many people here have asked someone else to be their mentor? Raise your hand. Ok, how many people have ever had someone ask them to be their mentor? How many people here have mentored someone else and enjoyed it? How many people would be open to mentoring someone but nobody's ever asked them? Find someone with more experience than you, and ask them if they'll talk to you over Skype once a month for 45 minutes. If they don't have time, ask them if they know someone who does.

Meet up with producers at other types of companies, and try to figure out what strengths you're overlooking. You're here at GDC at the Producer Bootcamp; it's a perfect place to do this! Find other producers here to go have coffee or lunch with; this may be the best opportunity you have all year!

- It takes less effort to improve a strength than to fix a weakness
- Ask your team what will take you from 'good' to 'great'

So which strengths should you build?

It's much more efficient and effective (not to mention enjoyable) to build on a strength you already have than to try to fix a weakness. Pick something you're already pretty good at, and get better at it. There are plenty of producers that I've worked with that I admire, but I'd never want to be just like them. And there are jobs they do amazing at but I'd suck at, and vice versa. I'm not going to try to build strengths I don't have at all: I'm never going to be a good salesperson, it's not something I want to get better at, and that time is better spent building my existing strengths.

Ask your team what would take you from 'good' to 'great'. Ask your team about the best producer they ever worked with, and what made that person so strong? (If it's you, awesome; work on improving whatever it is they think you're amazing at.) Your manager (if you even have one) is probably very busy, and if you're doing a good job it's not their highest priority to tell you how you could be even better. So ask: if you get a performance review, ask what would have made it better. If your team misses a milestone, ask what you could have done to prevent it or if there's something they wish you had done. Often, the easy answer is experience: If you had taken a battle royale game through Xbox cert before, you would have known about that one matchmaking queue requirement that we failed on. But sometimes you'll find a strength you can work on.



Picking Which Strengths to Build

- It takes less effort to improve a strength than to fix a weakness
- Ask your team what will take you from 'good' to 'great'
- Develop the right mix of specialized and general strengths for your role



For example, I was a level designer for 7 years before I became a producer. And I had to unlearn a lot of things: taking sides in arguments based on which feature I liked best instead of what was best for the project, for example. I had to learn how to speak for the project and for the constraints on it, instead of for my own design ideas. At times, honestly, this was keeping me from even being a good producer. But I certainly would have never become a great producer if my manager hadn't told me that I needed to learn how to be a neutral third party during debates and discussions. That was something I'd never really had to do, and now it's one of the most useful tools I have: being a neutral, dispassionate third party.

Balance general strengths and specialized ones. I worked with an FX artist who only wanted to do blood effects; he was really good at them; it was a strength, but it was way too specialized. Unless he happened to be working on a game with lots of blood, his strongest strength was basically useless.

Or, if your skills are all too general, then you won't be able to be effective in specific roles. Sometimes a team needs a producer who knows everything there is to know about Jira/Project integration. If you're that person, you have a big opportunity to have a lot of impact on your game's success.



Good & Great Producers: Redux

A **good** producer has solid producer strengths.

A **great** producer:

- Has the specific strengths their team needs
- Can match their strengths to their situation
- Knows how to build new strengths



So what's the answer to the question I asked at the beginning of the talk: what's the difference between a good producer and a great one? The more I ask this question, the more I think about it, the more I think the answer is: a good producer is someone who has some solid producer strengths. But a great producer is someone who's on a team that needs their particular strengths, who can match their strengths to their situation, and who knows how to build new strengths when they need to.

This is my own opinion, but if yours is different, please share in the Q&A!



Conclusion

- Producers make sure the game gets done
- Strength = Skill + Joy + Impact
- Different teams need producers with different strengths
- Focus on building and improving the strengths that will help your team



- Producers make sure the game gets done
- Your strengths are skills that you enjoy that have impact on your team
- Different teams need producers with different strengths
- And you should focus on building and improving the strengths that will help your team the most!

Thanks!

Thanks!



Questions?

- rtomandl@gmail.com
- [@slideruth](#)



Resources

- The Game Outcomes Project
- Google's Project Aristotle
- Alison Green's 'Ask a Manager' Blog
- The GDC Vault
 - Paul Tozour's 2016 talk
 - My scoping and prioritization talks
- Books:
 - The Goal
 - High Output Management

Email me for slides!

The Game Outcomes Project: <http://intelligenceengine.blogspot.com/2014/12/the-game-outcomes-project-part-1-best.html>

Google Project Aristotle: <https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html>

Alison Green's blog, "Ask a Manager": <http://www.askamanager.org/2011/12/how-to-be-awesome-at-work.html>

GDC Vault: <http://www.gdcvault.com/>

Especially Paul Tozour's talk: <https://www.gdcvault.com/play/1023258/The-Game-Outcomes-Project-How>

And my 2016 talk about scoping your game:

<http://www.gdcvault.com/play/1023138/Producer-Bootcamp-How-Saying-No>

My 2015 talk about prioritization, while we're at it:

<http://www.gdcvault.com/play/1022221/Producer-Bootcamp-How-to-Prioritize>

The Goal: <http://www.amazon.com/Goal-Process-Ongoing-Improvement/dp/0884271951>

High Output Management: <https://www.amazon.com/High-Output-Management-Andrew-Grove-ebook/dp/B015VACHOK/>



Takeaways

- Strength = Skill + Joy + Impact
- Match your skills to your team
- Talk to Producers with different strengths
- Take Project Management classes
- Improve strengths, not weaknesses
- Default to 'Yes' for trying new things
- Vary your experience
- Ask your team for feedback



Takeaways:

- Strength = Skill + Joy + Impact
- Match your skills to your team
- Talk to Producers with different strengths
- Take Project Management classes
- Improve strengths, not weaknesses
- Default to 'Yes' for trying new things
- Vary your experience
- Ask your team for feedback