

Welcome to DESN 1145 - Game Management 1

What's up with your projects?

Project management

Project mgmt is actually managing all the resources needed to complete a project on time and on budget

Project mgmt is more black and white when there are clear product requirements and those are all that matter.

In games there is one significant factor that is very hard to plan: fun. Also: project manager = producer

So in games we have: pre-production (less planned - all about **“finding the fun”**) and production (more planned - once the fun has been found)

You need to keep in mind the following:

- people (skills, available hours, dependencies, holidays, vacations, other commitments)
- hardware (dev kits, PCs, servers, drawing tablets etc...)
- software (# of licenses, pipeline, versions)
- recruitment time (employees and contractors)

Typical Milestones

- **Prototype** - showcasing the fun
- **GDD**
- **Alpha** - all main features, not all content
- **Beta** - all features and content but unbalanced and buggy
- **Release candidate (RC)** - should be all done but still gets more QA
- **Gold** - done

Other:

Vertical slice - typically a fully polished level (RC of a level)

Demo - a polished piece showing off a particular mechanic or area in the game

Marketing assets: concept art, mocks, screenshots, teaser trailer, trailer(s), gameplay video(s)

Making it all work

You don't always need all of the things on the previous slide

But it helps to know WHAT YOU WILL NEED before you start.

Some dates can be changed, others (**like LEVEL UP**) can't

Once you've figured all this out - it's time to make a gantt chart

Stuff for Level-Up

Game demos/prototypes/vertical slices

Trailers (a week or so before the event)

Studio logo

T-shirts

Biz cards

Misc - buttons, stickers etc...

Create dates for when you need to order / start making these things to have them ready on time

And now a bit of history

In the good old days the end customer only saw the final game (alpha, beta and release candidates were not public)

Then more and more games started inviting players to their betas. Brilliant marketing - making game testing sound like something players should desire to do...

Now devs give access to alphas via things like kickstarter (and various other self-starters). Minecraft was the poster child for the early access thing.

It varies by platform. Apple does not allow anything like a beta. PC is a free-for-all. Consoles sometimes have beta programs.

... a bit of history - continued

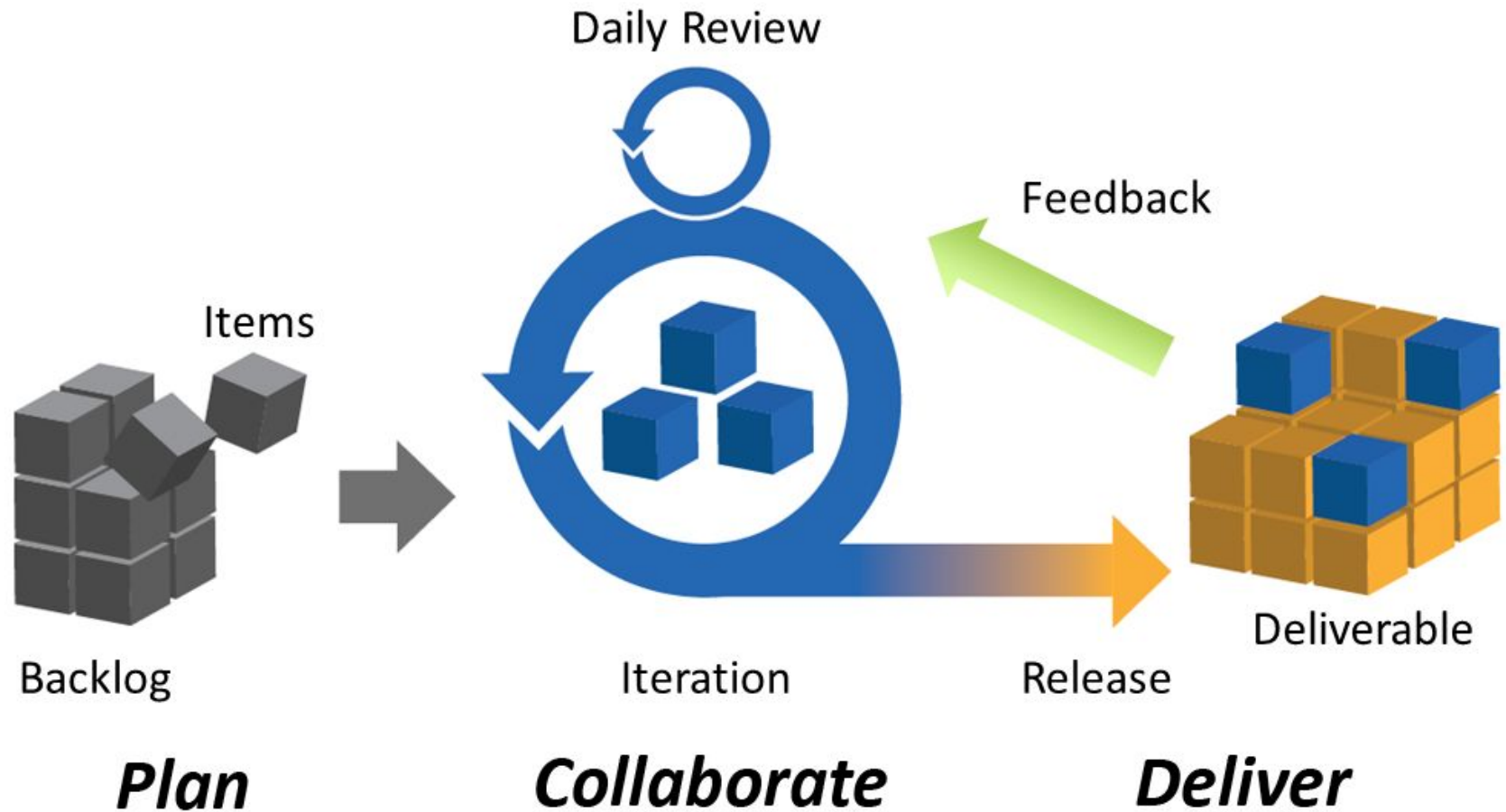
Back when work-for-hire was the predominant model for making games the alpha/beta/RC were much more popular. (*why?*)

Now indies do whatever they want which results in a lot less definable milestones.

Development methodology also had an impact. Project planning was a lot easier in a waterfall model. Agile makes things more complicated.

Because of DLC, updates, and games-as-a-service many projects have no end.

Agile Project Management



The Scrum Board



Backlog: **features**

TO DO, In Progress, Evaluate, Done: **tasks**

Trello - Use It

←

Boards

🔍

Trello

+

🕒

🔔

WK

Wojtek's test project

☆

Personal

🔒 Private

Show Menu

Backlog (features)

Player Vehicle

Enemy Vehicle

Prototype level

Menus

HUD

Add a card...

To Do (tasks)

Texture the player vehicle

Rig the player vehicle

Animate the player vehicle

Add a card...

In Progress (tasks)

Model the player vehicle

Add a card...

Evaluate (tasks)

Add a card...

Done (tasks)

Create concept art of player vehicle

Find reference art for player vehicle

Design player vehicle

Add a card...

How do you make all this happen?

Use your team structure.

Designer explains the game to producer, tech lead and art lead

They agree on a way of breaking the game down into meaningful 'chunks'

Each lead sits down with his team, breaks the 'chunks' down into tasks and estimates how long tasks will take to complete

All of that goes back to the producer who together with the designer and the leads puts together a project plan

Gantt Charts

Old saying:

“Planning is more important than the plan”

VERY true for games.

Project plan is a tool for:

1. tracking how far behind you're falling so that you can make adjustments to the project
2. making sure that people do things in the right order

If you use it as a whip people will lie and you won't know where your project is at.

Gantt Charts cont'd

A project plan is also useful for:
BUDGETING a project

A couple of tips:

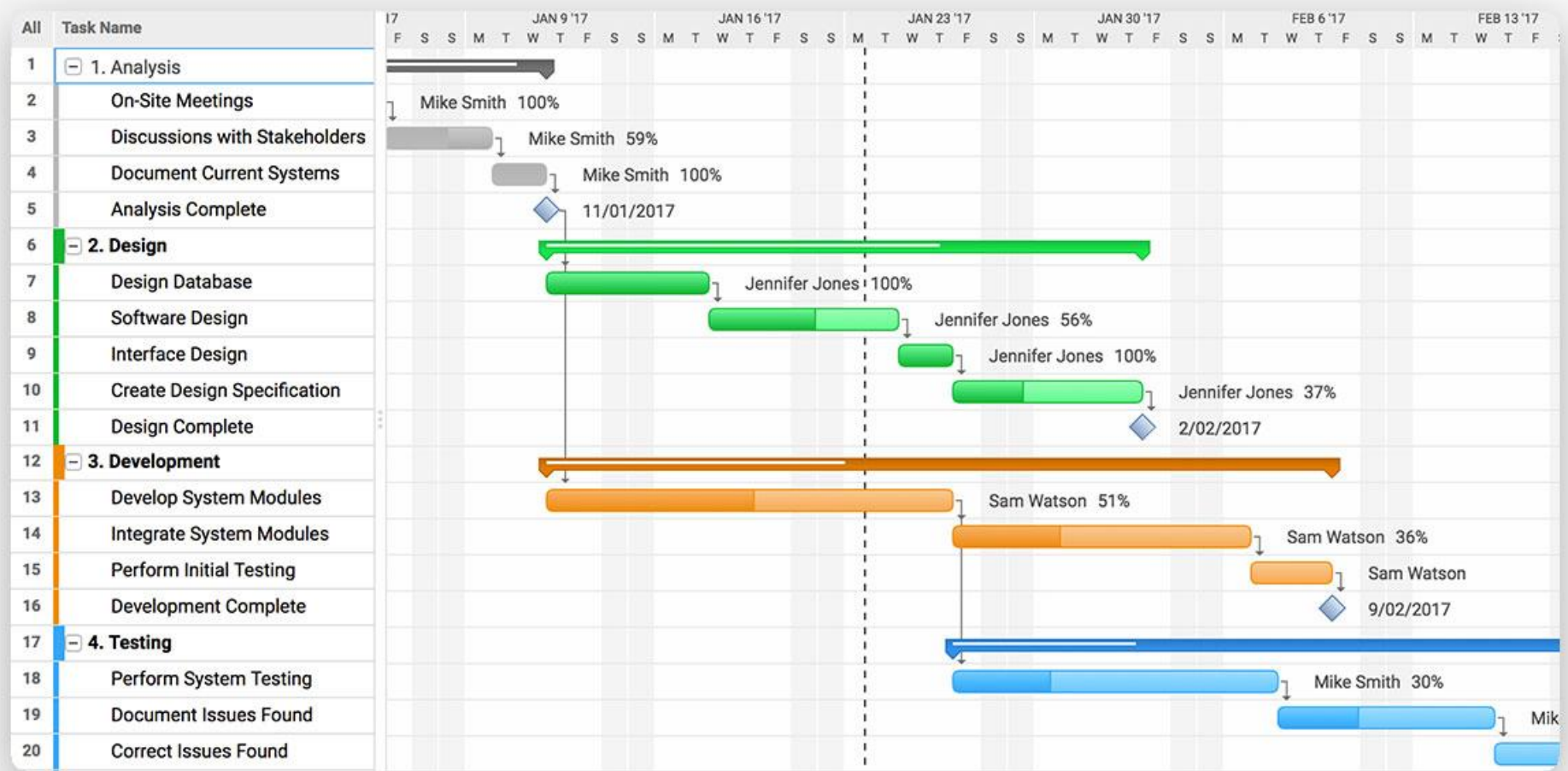
- don't believe anybody's estimates (double everything)
- if you're doing contract work add a profit margin
- remember to add non-labour costs

Gantt Charts

There are many project planning tools

- MS Project (big - usually overkill, expensive)
- Merlin (Mac only - not bad but still have to pay)
- Ganttter (formerly free, online, part of Google Drive, very light but quirky)
- Agantty - it's basic but it's free. Also it starts off being German :-) I recommend trying it - it's likely easier than Excel

Example Gantt Chart



You can use EXCEL!

	A	B	C	D	E	F	G	H
1	OUR AWESOME GAME PROTOTYPE PLAN							
2				Oct 1 - 7	Oct 8 - 14	Oct 15 - 21	Oct 22 - 28	Oct 29 - Nov 4
3	Main Menu							
4		Background Art						
5		Decide on what it should be	designer					
6		Make the art asset		concept artist				
7		Implement the art assets					coder	
8		Sound						
9		Decide on the type of music		designer				
10		Compose the music loop			composer			
11		Implement the music loop						coder

Organizing tasks

There are a couple of ways to organize things

1. By functional group

- dev team
- art team
- design team
- etc...

2. By feature

- menus
- walk cycle
- etc...

**In all cases line things
up to your milestones**

Bug Systems

Not just for bugs - they are task systems

- they are a tool to assign tasks
- they are a way of prioritizing work
- they are a way of ensuring particular features are done for appropriate milestones
- they are a way of keeping resources accountable
- you can use them for both code and art
- good idea to have your GDD first, then your project plan, then start making tasks

In the last few years tools like Trello have largely replaced bug systems as task tools

Workflow

You need to create a flow/process that you'll use for your tasks

- who can create tasks (what kind)?
- who can modify them?
- who can approve them?
- who can close them?
- do you want to do code review?

Let's figure this stuff out before you get too deep into your projects and before you start using a task system

Some examples

Trello - USE IT

Jira - like Trello but you have to pay (it does have extra features)

Redmine - free and flexible but you need to set it up.
Includes a wiki

Bugzilla - similar to Redmine but more focused on just bugs + not sure if it comes with a wiki

FogBugz - cheap but not free - good option if you don't want too much set-up

Here's a full list:

http://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems

It's good to have a place that can act as your knowledge base.

You may want to keep a list of tech standards, file naming conventions, perhaps the GDD + other info that you want the team (clueless Winterns) to have access to

Some bug systems come with wikis but you can also find stand-alone ones like google sites

This is not a must but it can come in handy especially if you use it for your GDD

Code repositories (source control) - a must

Asset repositories - nice to have

Used to keep a single 'current' copy of code/asset, track change history, allow multiple people to code concurrently.

Code repository examples: GitHub, Mercurial, Subversion. Some can be set-up to be used for other assets

Dropbox

It's good to have a place to share files especially if you don't use an asset repository.

If you're good, you can use Dropbox (or something like it) as part of your art pipeline.

Watch this talk by Get Set Games

http://www.youtube.com/watch?v=76EY_7PzWUM&feature=youtu.be

There are others so do some research if this sounds interesting. You'll probably use Google Drive :-)

I recommend this (I think you're using it already)

You can use it as an asset repository

You can sync it to your computer

Please use something like this and NOT FB messenger or Slack or Discord to pass assets around. Artists will want to do that. DON'T LET THEM.