



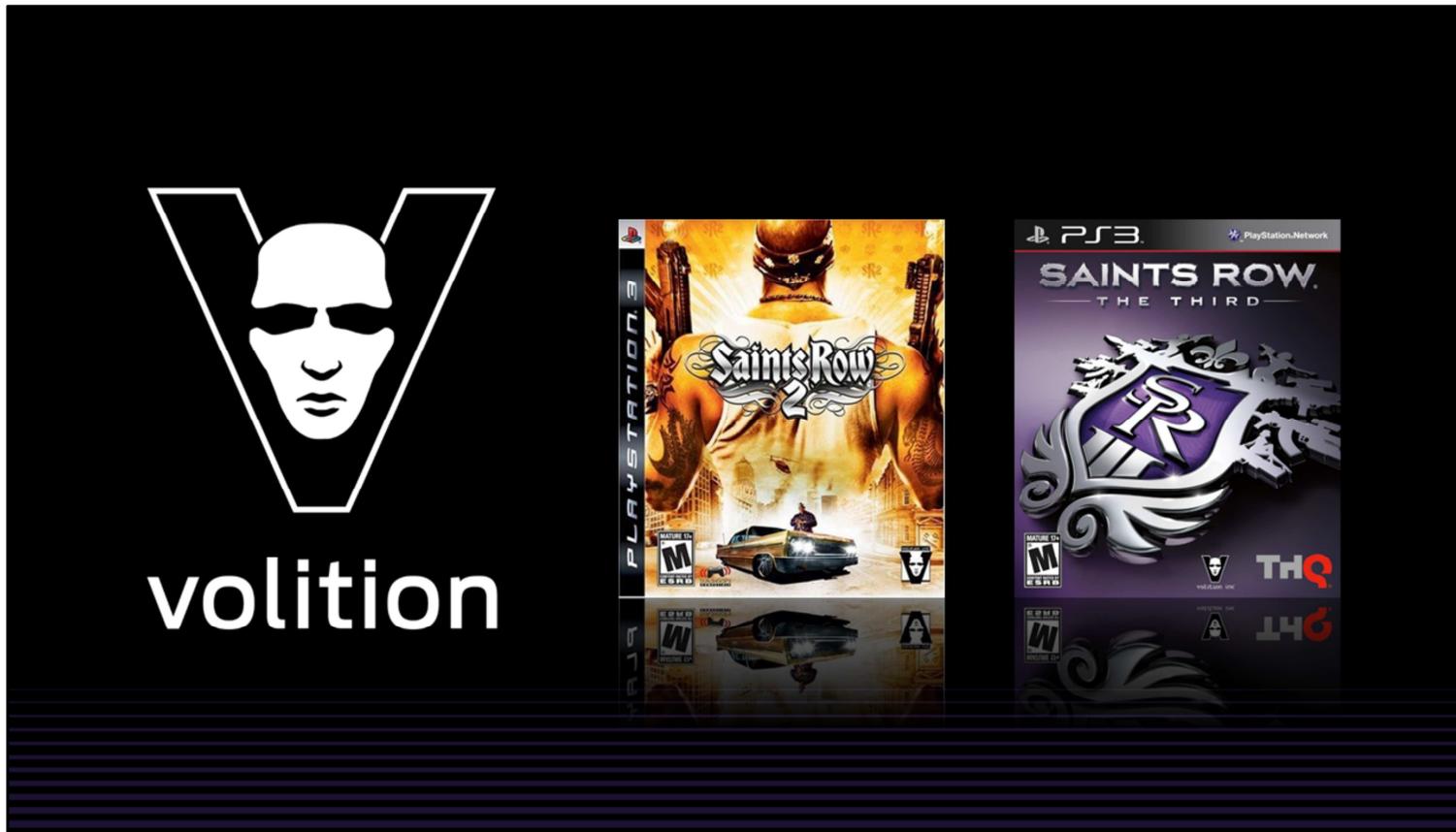
volition

THE VERTICAL SLICE CHALLENGE

Greg Donovan
Senior Producer

General Welcome

Let's get going.



As way of a brief introduction:

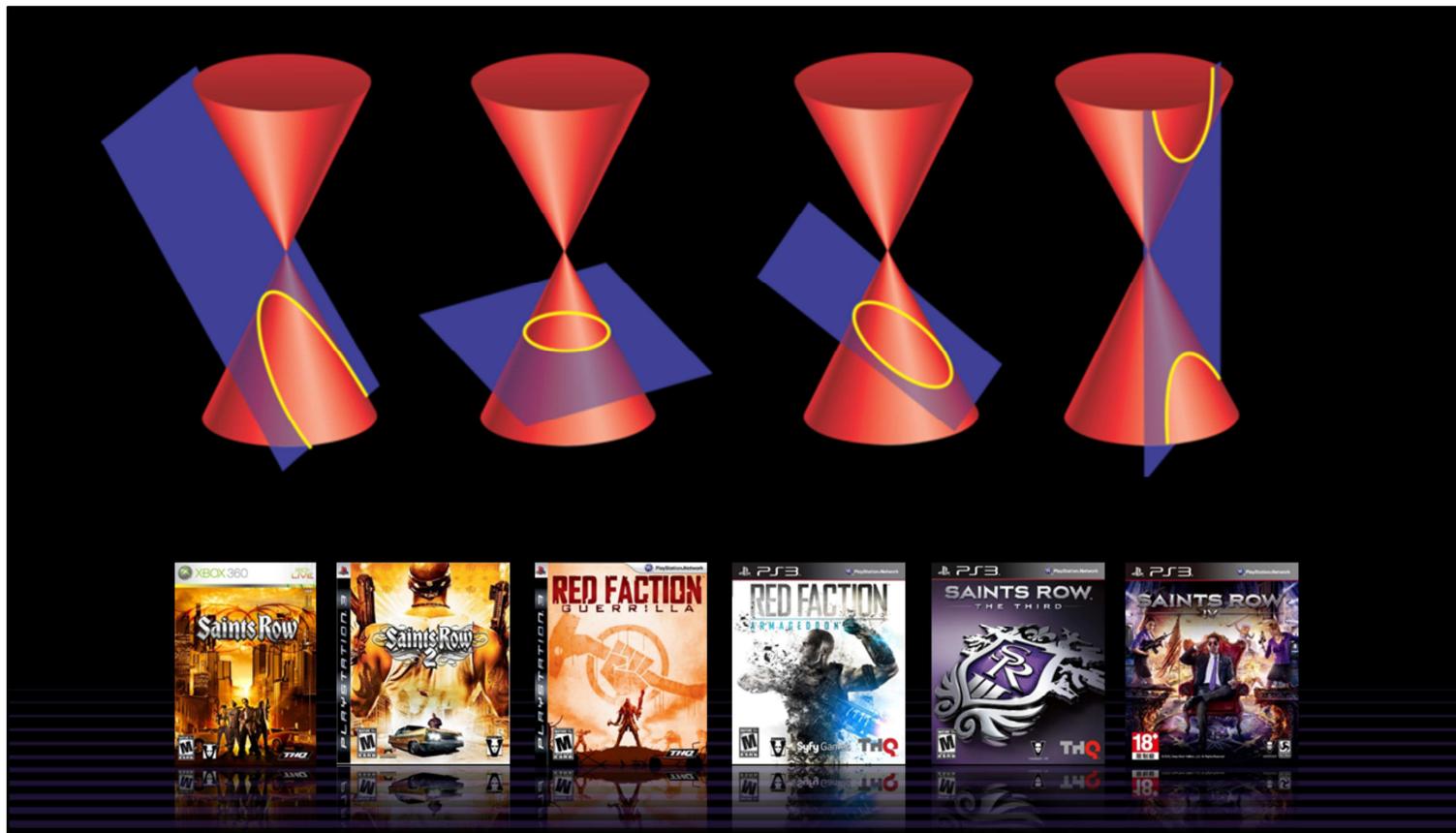
Senior Producer at Volition.

Volition has been putting up with me for a decade
16 years in the industry

CLICK

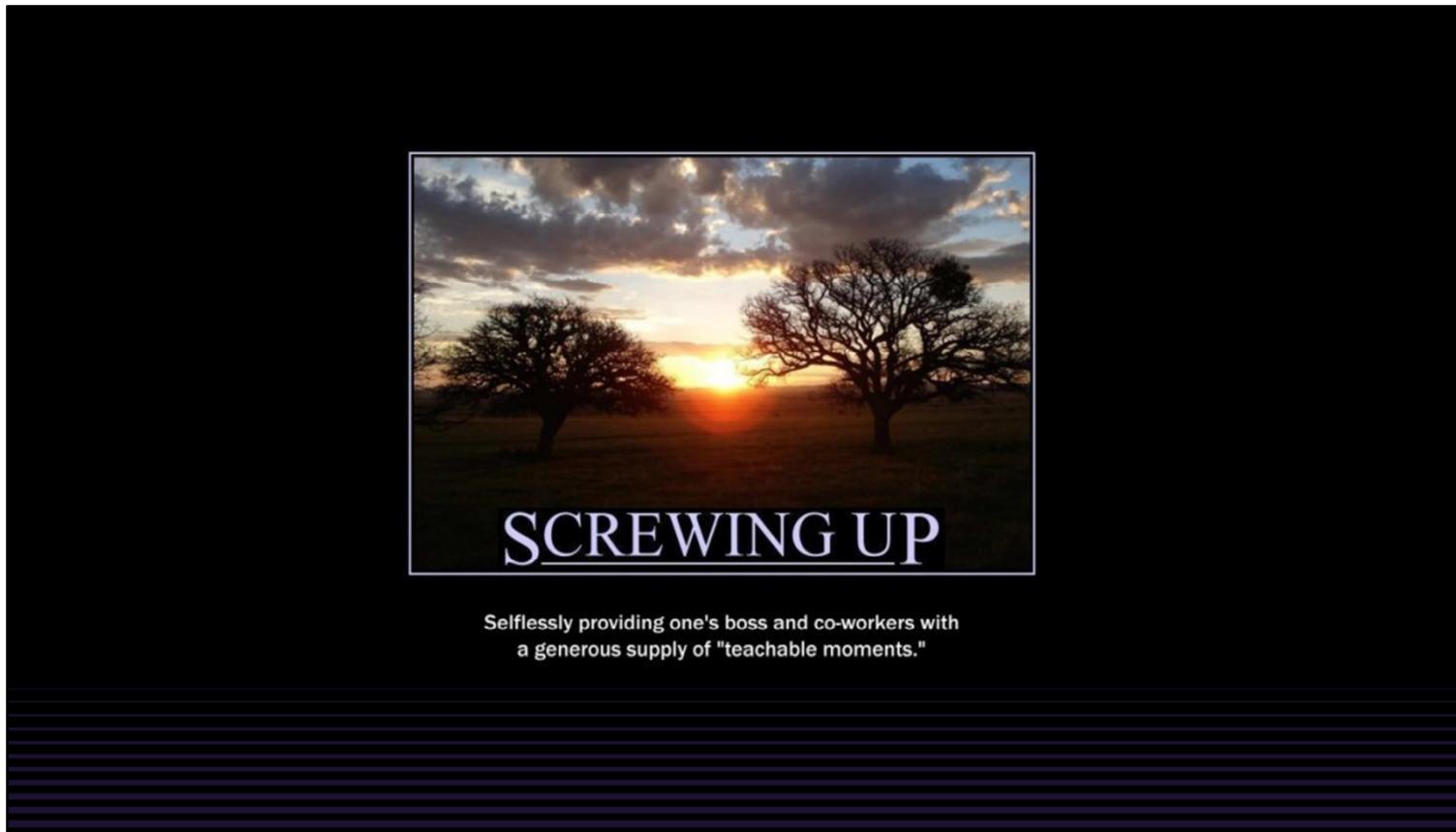
- Lead producer on SR2, SRTT,

TALK SLOWER!



This talk is about:

- The challenge of delivering on a game's vertical slice
- And show Volition's history of Vertical Slices through examples from the Saints Row and Red Faction franchises,
- We'll review the successes and failures of this deliverable on each of these titles (sneak peek: we've had more failures than success... but that's okay!)
- Volition believes the Vertical Slice should be part of the development cycle because even with its challenges it can really help the team before entering production.
- This talk aims to illustrate this point and show why.

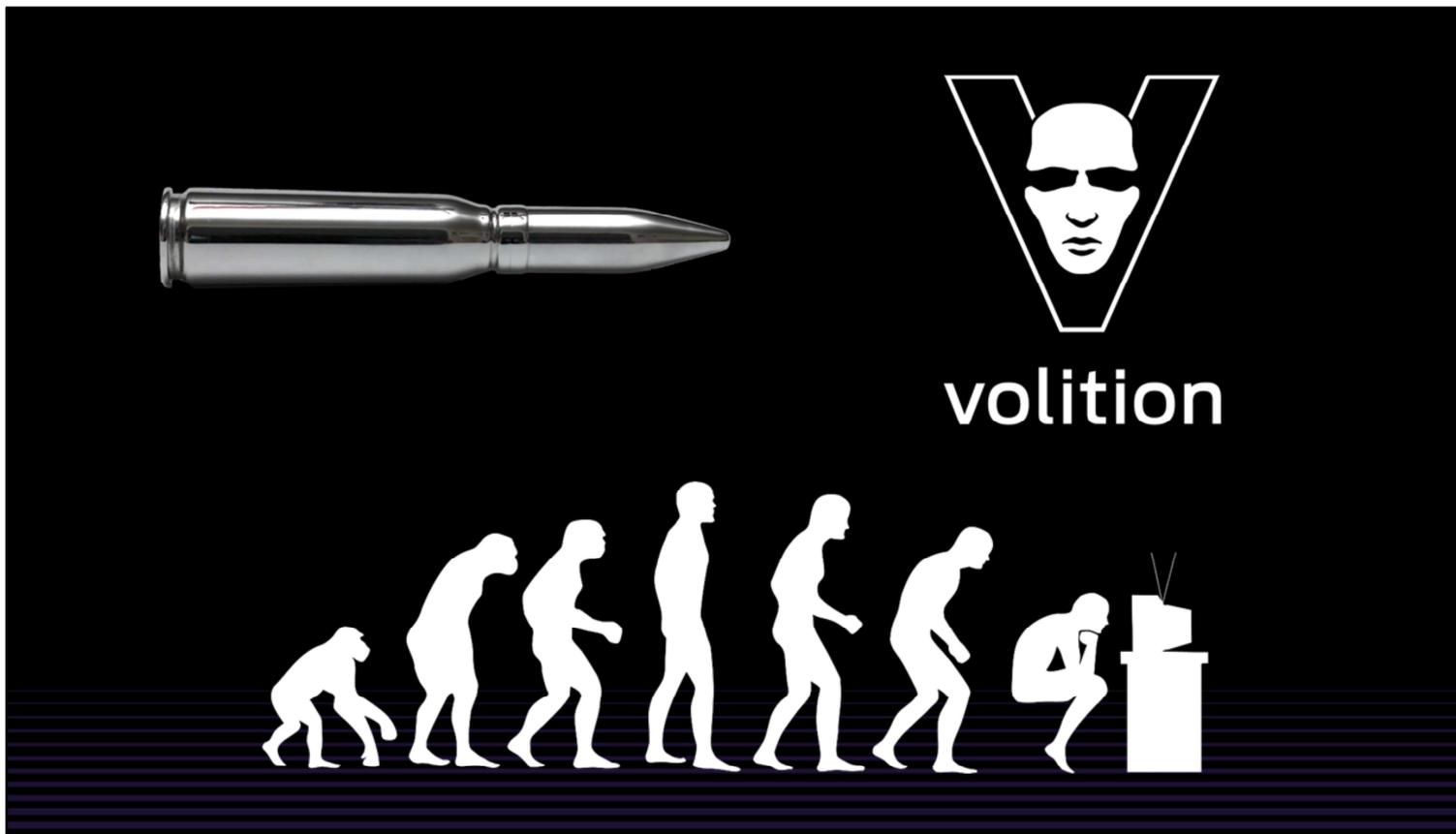


We're giving this talk (hopefully) to
Educate, share and inform – to potentially help you and your teams avoid the mistakes we have in the past.

Because we believe that sharing each others mistakes and learning from them means better games, and that's something we all want.



But first there are some caveats about this talk.
And you need to know them.



Caveats:

ONE

This is not a “do this and you will have guaranteed success” lecture. We’re not claiming to have THE ANSWER. I’ve been to lectures like that and I think they’re horseshit. Think about it - We (production) basically try to accurately schedule profitable fun, based on a process that is inherently iterated upon again and again. This is really, really hard. So if you’re looking for a silver bullet from this talk, you’re going to be really disappointed with me.

TWO

This talk is: Where Volition is now based on our past learnings.

How we approach the Vertical Slice today may be different on future projects

Moreover, this is based on what we do, and what works for our studio (we have about 200 FTE’s and make largish, open-world games on the console and PC); What we do may not be right for you

THREE

This talk isn't about me. I didn't work on all the projects. This is about the studio and its learnings; I'm just a mouthpiece. I did do a lot of interviews, though, with many principals on the projects I didn't work on, to ensure accuracy.

What is a Vertical Slice?

So, let's spend a few moments answering this
(apologies if this is insulting – I understand that most of you probably know what a Vertical Slice is)
But I'm doing this to ensure we're on the same page, and are using a common language for this talk

First Playable

Proof of Concept

Minimum Viable Product

$$\begin{aligned} -1 &= -1 \\ -1/1 &= -1/1 \\ -1/1 &= 1/-1 \\ \sqrt{-1/1} &= \sqrt{1/-1} \\ i/1 &= 1/i \\ i &= 1/i \\ i^2 &= 1 \\ -1 &= 1 \end{aligned}$$

Vertical Slice goes by many names and labels

Sometimes: First Playable

Sometimes: Proof of Concept

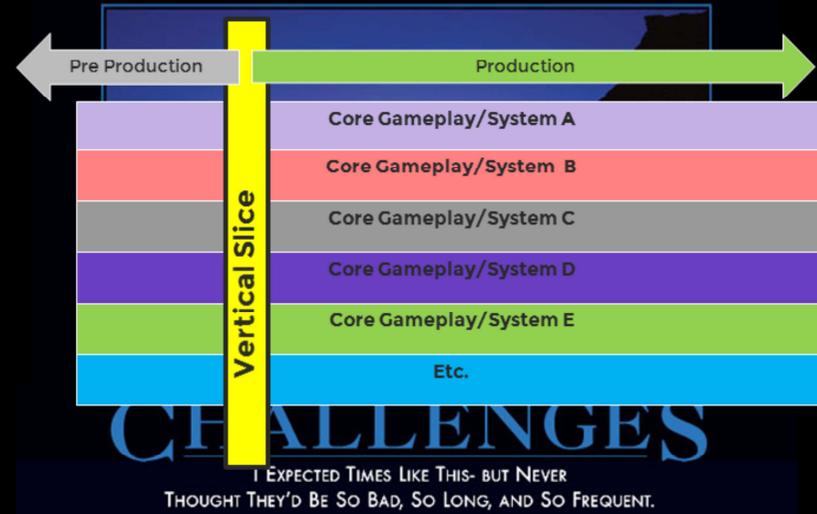
Sometimes (if you're from silicon valley) a MVP: Minimum Viable Product

Whatever. All the same thing, essentially.

But, for common understanding:

THE VERTICAL SLICE

- Not a full game
- Player Experience Focus
- Major systems
- Key Features
- Representative of final quality



The Vertical Slice is

A section (level, gameplay duration, X number of missions, whatever) of the full game that successfully communicates the team's intended PLAYER EXPERIENCE by having major systems and features proved out in preproduction, working together. Assets are typically not final, but they are or should be – as best as possible - Representative of Final Quality

Yes – I made that lovely image.

CLICK

Generally speaking, the Vertical Slice is a very largish and complex milestone/deliverable that occurs just before production

They are challenging deliverables

But we believe they are worth the effort BECAUSE they can teach you a lot.

Not least of which is...

GATE TO PRODUCTION

- Player Experience
 - Core Gameplay
 - Critical Features
- Technical Debt
- Objective & Honest

Technical Debt



Customer's view



Developer's view

Whether or not the team is ready to move into production.

As a team you can determine if you're ready to move into production - Where Production is loosely defined as: The team knows what they are making and how to make it -

By attempting to (objectively) answer these questions:

CLICK

Is the game delivering on intended player experience? Do you get a sense of what the game is?

CLICK

Do you have an acceptable level of Technical Debt?

Quick aside for Technical Debt:

No way is everything "done" at the Vertical Slice. Some features and systems will be hacked in. Others will still be in

development. And there may be some that haven't even been touched yet.

The issue is – the team will need to finish these systems at some point, or otherwise architect them correctly.

The rub: Quantifying and tracking how much work is left to bring these non-finished systems and features to a level where they are considered done enough to submit

CLICK

Succinctly: The vertical slice deliverable can keep the team honest about progress and whether or not you're ready to move into production (if you're objective)

Because overall and importantly- The issues you uncover in a vertical slice (and you will uncover some) are best discovered then instead of the heart of production.



So, let's go down memory lane and take a look at Volition's attempts at Vertical Slice milestones: what worked, what didn't and – using the wonderful power of hindsight 20/20 – what the key lessons from each was.

Before 6:00

5:30 is better



First up (going on chronological order here) – Saints Row
Xbox 360
Late August 2006

THE HISTORY

- New Genre
- New Tech/Tools
- New Platform



I used a similar slide last year, but it still applies for this talk:

Saints Row was really, really hard for Volition.

Open World – new genre for the studio

New Tech and Tools

New Platform (Xbox 360)

WE DIDN'T KNOW HOW TO DO MUCH OF WHAT WE WERE DOING, WELL AT ALL

CLICK

Basically: Saints Row 1 development was shaping up to be a perfect storm.

BUT: Saints Row marks the first time Volition consciously planned for a Vertical Slice

THE HISTORY



Before SR1, team sizes at Volition were ~50 – 60 people. And we didn't do vertical slices. In part because the concept wasn't a thing at the studio and also because with a team of 50 – 60 it's (relatively) easier to know what each other is working on. Things just came together. It was cowboy development.

But with SR1, that approach wouldn't work.

CLICK

This is the producer on Saints Row 1, Jacques.

I don't have any real reason to place his image here except to troll him a bit.

Anyway, he and some others in studio management worked at companies where Vertical Slices were a thing.

They, being smart fellows, introduced the concept to the team.

And Volition, also being of relatively smart fellows, saw the benefit of the concept.

Accepted; not a dumb idea

So, they came up with a plan for the Vertical Slice deliverable:

THE PLAN

- Phase Based
- Player Experience
- City Section
- Core Gameplay
- Sell the Dream!



The Plan:

- Make this deliverable phased based (multiple milestones across multiple months). This was going to be a lot of work (NEW EVERYTHING) with lots of iteration; it was going to take some time

CLICK

And this deliverable should culminate with:

- A demonstration of the player experience, playable in

CLICK

- A small section of the city that focuses on core gameplay

CLICK

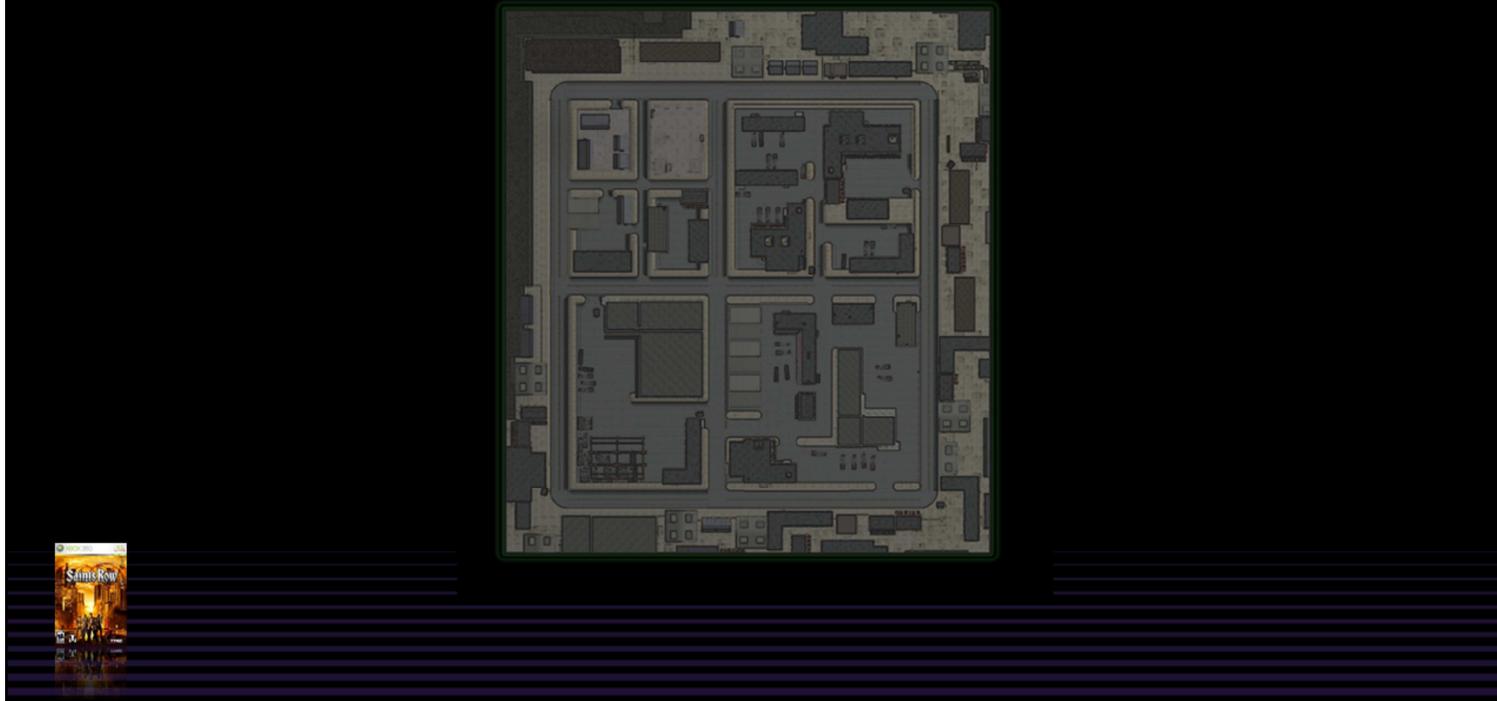
- And overall show progress and sell the vision to both the publisher and the team
- Internally we needed to show we could make Open World Games

- And for the publisher we needed to show the actual vision and tone in something playable game as Saints Row was viewed by some at THQ as very controversial game)

CLICK!

Anyway, Onward! A plan was set, the team was feeling good about prospects, and they headed into the storm with confidence.

THE RESULT



Here is a map area for for Saints Row's vertical slice.

It was ultimately called "Phase 5" because that's how many phases it took to get it to the intended deliverable.

As you can see – it's one section of the city. Approximately 4 blocks.

Next slide is a video clip of gameplay footage is from Phase 5.

Full disclosure: This clip is actually one month after the last documentation I could find on Phase 5, but everyone I spoke with on the Saints Row 1 agrees this footage is from the Vertical slice deliverable.



I'd like to point out that that rather brisk fps was not altered in any way.

10:00 or earlier

9:30 is the target

THE GOOD

- First Player Experience
- Full City Block
- Scope coming into focus
- Sold the Experience



So, looking back at SR1, here's what went well:

- Demonstrated the Player Experience and gameplay through core features working together: locomotion, combat, navigation, vehicles, etc.
- The fun was there

CLICK

- The deliverable was a contained city section. The team had a first pass idea of what it would take to make a city

CLICK

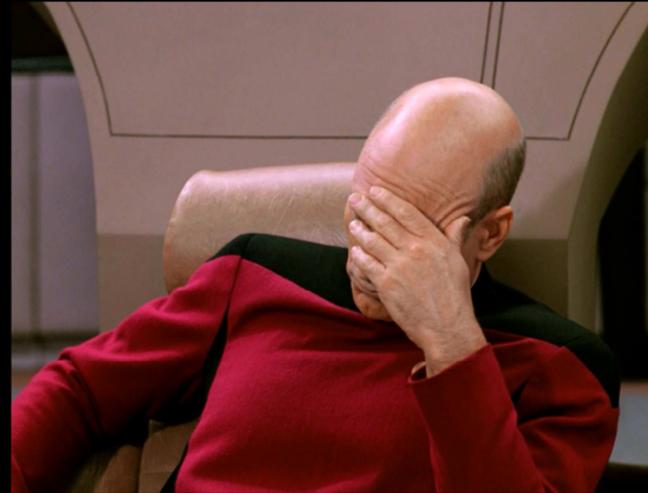
- Answered some good questions for the dev team:
 - Scope was coming into focus: It was going to be huge!
(In fact, this deliverable started conversations about pushing the date)
 - Seriously – knowing that you have a scope or timing issue as early as possible is a good thing, and a VS can help add “data” to otherwise subjective concerns about a date

CLICK

And finally: This deliverable got THQ and the team excited, precisely because a paper vision was actually in game and you could PLAY IT

THE BAD

- Scope was HUGE
- Tech Debt OMG!
 - Streaming



As for the things that didn't go so well:

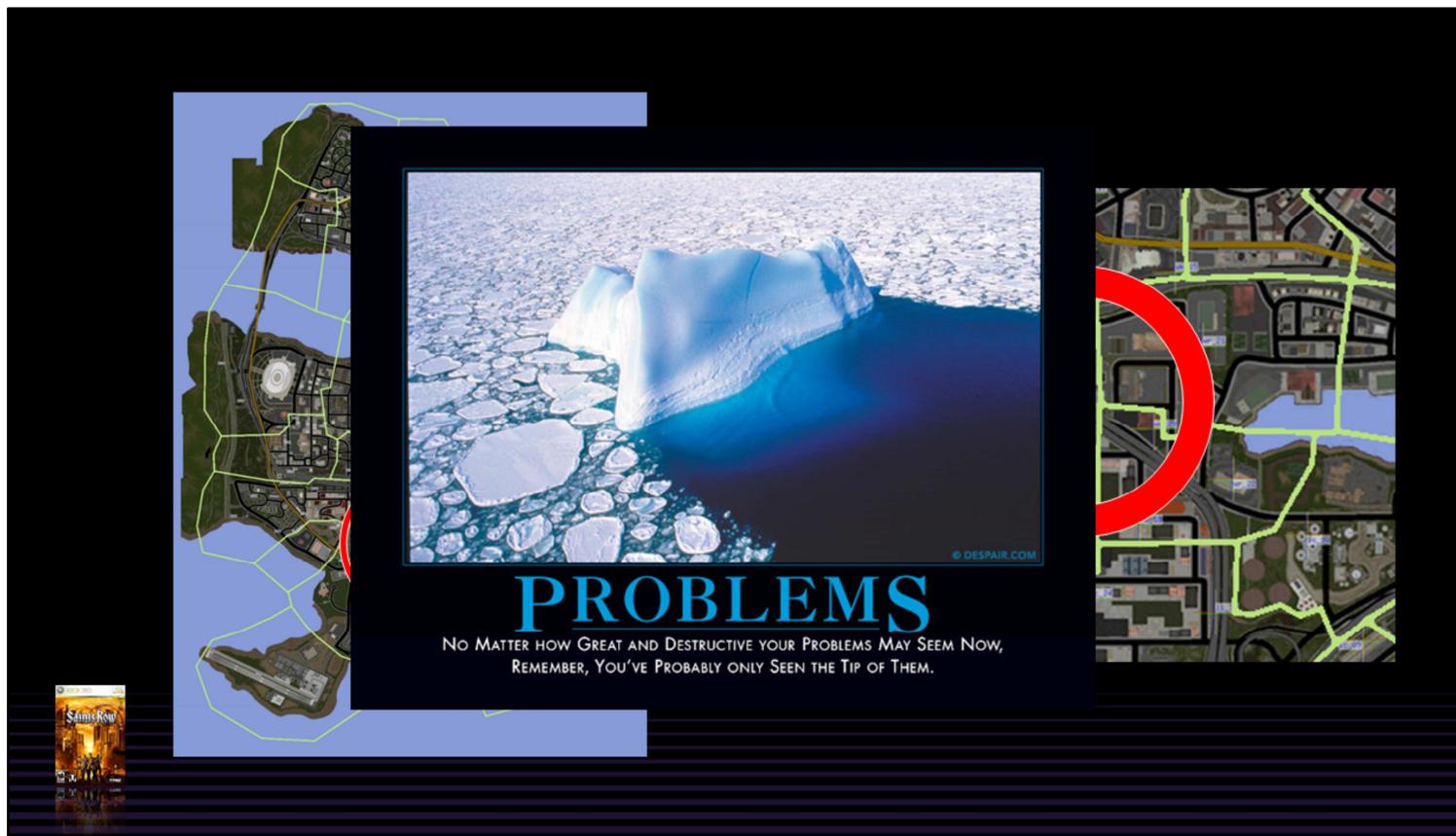
- Scope was huge. Remember – the date was in questions, at least internally, based on this deliverable. Instead of looking for ways to work smarter (like reducing scope), we told the team to work more.
- SR1 was very bad crunch for most the studio. In retrospect we could have done things a lot better.

CLICK

•Tech Debt wasn't understood or quantified. This was our first attempt at making an open world game. We had no idea what we were getting into precisely because there were so many new things.

•And although there were many unproven technical issues (it's never ideal to make the game and tech simultaneously, but sometimes that can't be helped) we'll focus on streaming as a specific example.

Streaming – as a system for first timers – is a really big matzo ball



To help provide some context here.

Streaming is rather critical for open world games. You can't have everything in the world loaded in at once, like you can with many linear, level-based games. Objects, animations, etc. all the DATA streams in around the player (and you do your best to hide it). If your streaming isn't efficient, you can't run the game.

On the left here is a map of the SR1 world. It's divided into city chunks. At the time, we had a chunk-based streaming engine. And it could only have two chunks loaded at once. The problem was (and we didn't fully realize it at the time), there were a number of areas in the world call TRI POINTS. Basically – a point where three chunks met.

CLICK

Like this. Whenever a player came across these areas, particularly in a vehicle where they were moving relatively fast, the world would basically crap out and things would disappear and slow down.

Let me blow up this area of the map. So you can see it better.

CLICK

Again – Phase 5 was basically two chunks made with an incomplete streaming system. We had no idea that this tri point problem would be an issue.

Basically we let the team go wide after Phase 5 and the chunk sizes – instead of being uniform and in a brick pattern – were all arbitrarily sized, dictated by world artists.

CLICK

Throw in the fact we hadn't even yet touched LOD's (level of details – phase 5 had NONE) and things like shaders – which are more or less common terms these days - were completely new to most of the industry at the time save for some true engine nerds... basically the tech debt assumed on SR1 was a monster that bit the team hard.

THE LESSONS

- **Vertical Slice is helpful**
 - Identifies Issues
 - Helps sell the Vision
 - Doesn't identify everything
 - Phase-based works
 - Open World Games are hard



HINDSIGHT
Decisions That Look Really Good At The Time
Could In The End Make You Look Really Foolish

As for the lessons (looking back with the benefit of Hindsight 20/20)

Overall Phase 5 was deemed a helpful deliverable. This should be a thing for us. And part of our normal development schedule.

CLICK

- It Identified issues the team would have otherwise missed
- Helped sell the vision
 - It's good to have a playable demo that communicates the vision and proves out gameplay before production
- Doesn't identify all issues (Certainly we didn't realize how much technical debt we had amassed)
- Phase Based is a good thing (Lots of work). This monster cannot be a typical one month milestone.
- Making open world games is hard.

FUN FACTS

- Musical
- Potential Visibility Set (PVS)
- 3DS Max



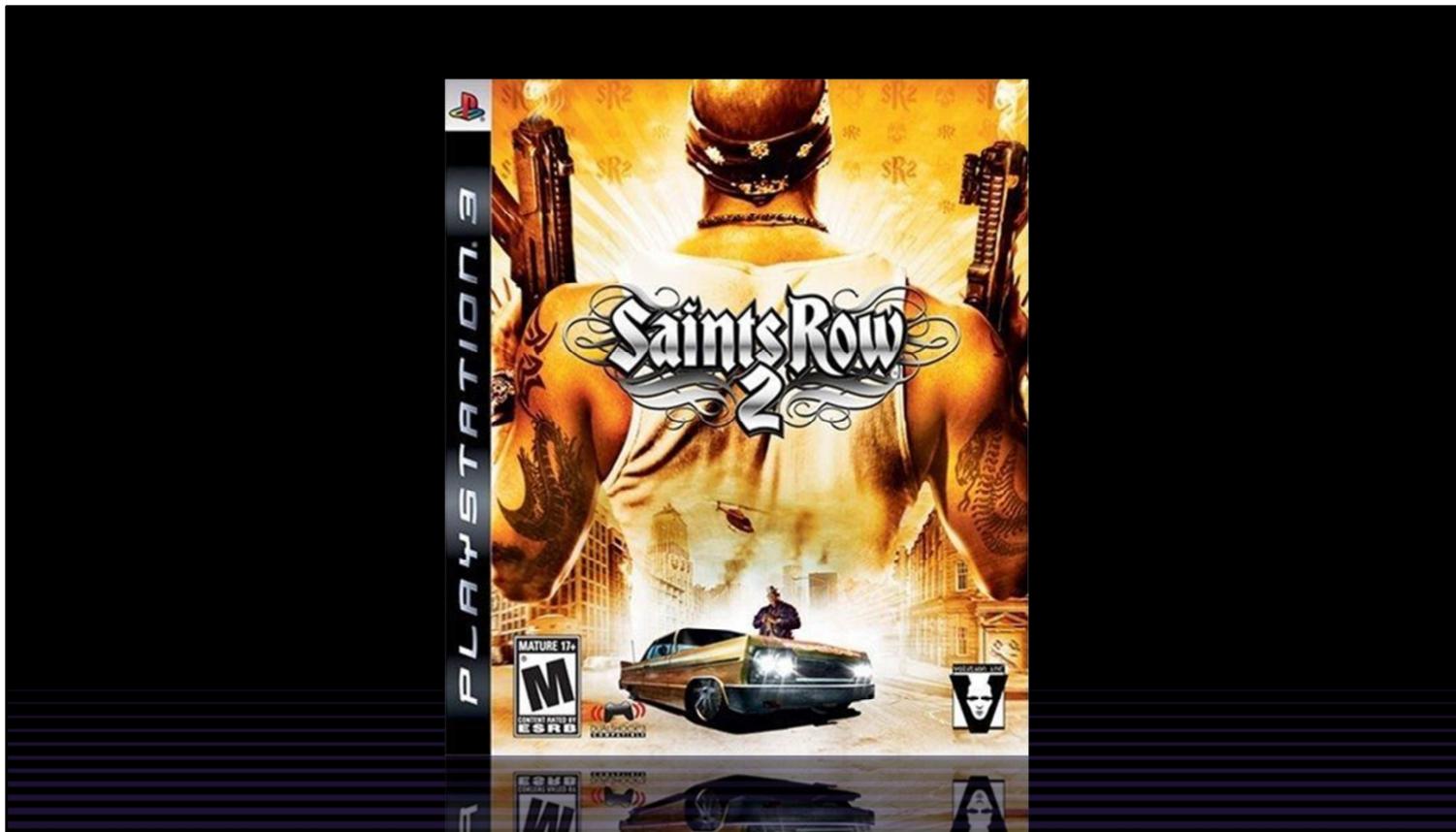
-After SR1 launched, a witty youtuber composed a musical entitled “Buggy Saints Row”. A musical. And although it was very painful to watch when it came out, we can (mostly) laugh at it these days. It’s still on youtube. Looking back, we have mixed feelings about creating a game that inspired someone to compose a musical, even if it was about performance issues.

-SR1 was running at ~10 – 15 fps three weeks before submission. Failure to establish tech and budgets earlier in production was likely the largest contributor to this. The hail mary was PVS – Potential Visibility Set – basically a giant rendering farm used to calculate performance. Programming saved us. Epitome of “necessity is the mother of invention”

- The SR1 city was modeled and created in 3DS MAX.

Think about that. Max is a great authoring tool, but a world editor it is not...

Okay – moving on to other titles...



Saints Row 2
October - 2008
Xbox 360, PS3, PC

16:30 or before

THE HISTORY

- Doubling Down
- New Leads Team



Saints Row 2

Preproduction on SR2 started well before SR1, we were banking that SR1 would be a hit. We necessarily we had an entirely new leads team...

CLICK

This new leads team (of which I was a part) was... confident.

From the outside looking in, it's easy to do things right. Certainly there would be no problems with us in charge. How hard could this be?

The SR1 team was over-worked and stressed. And they were finalizing. They didn't have time to work with us. And because of our attitude, nor did they want to.

So... we did things our way.

And that way... was:

THE PLAN

- New Features!



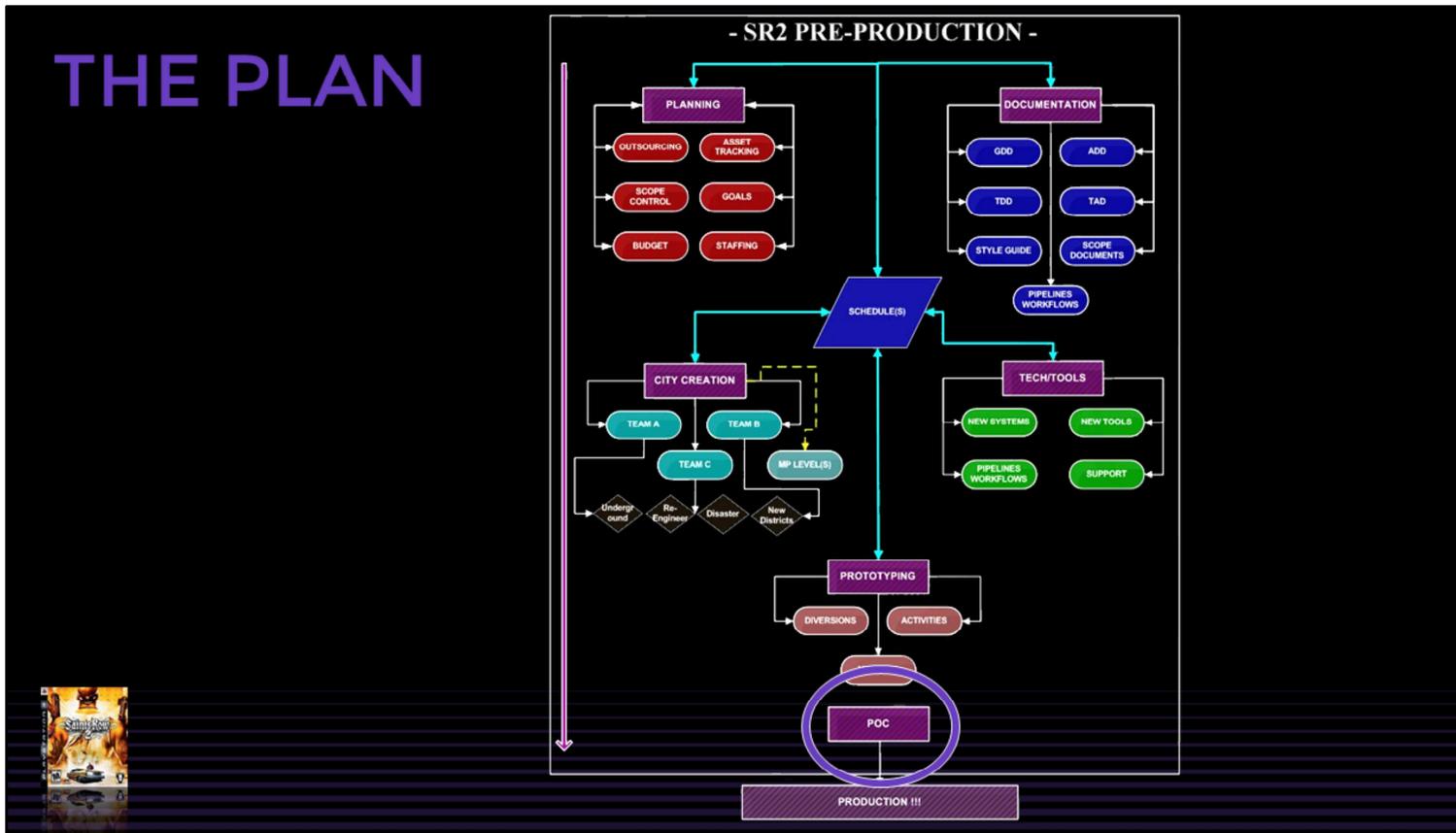
Take the code base and implement new features:

Features the SR1 team wanted but didn't have time to implement
Features the publisher wanted

(and most importantly, of course)
Features the SR2 team wanted

And our plan was to make a Vertical Slice that demonstrated these features in all of their glory.

THE PLAN



So, going through the archives, we couldn't find a build of the actual SR2 vertical slice.

But we made one. We made something. I found this documentation in the archives.

Lovely visio file that shows we aspired to have a Vertical Slice (we were calling it a Proof Of Concept) done before moving into production.

So either we deleted it or lost it in the archives. Whatever.

We did find a video called "SR2 Vision Piece". And that's what is shown on the next slide. And through talking with some of the principals on SR2, we agreed that most of the gameplay depicted in this video would have been in our Vertical Slice (based on other documentation we found).



Before 18:30

THE GOOD

- New Features
- Excited Team
- Excited Publisher



Right, so What worked:

- We got in a lot of cool features. Many of the features players expected to have in an open world game that SR1 didn't: boats, motor cycles, flying.
- And improvements to some of SR1's core gameplay, such as improved character creation and customization.
- In this regard, we were nailing the sequel formula – take the established core of the predecessor and improve upon it. Certainly by adding more and improving on existing features is one approach to making a sequel.

CLICK

- And as with SR1, this video helped solidify the vision of the game to both the publisher and team.

THE BAD

- No Player Experience
- Tech Debt



What didn't:

- We took a shot-gun feature approach. There was little to no focus on establishing a cohesive player experience
 - (SR1 focused on Player Experience – even if it was Thuglife or an MTV rap video, it was a defined experience)
- This hurt us in production in that we had a lot of people developing wide instead of deep.
- SR2 tonally is a weird game: In one mission you can be euthanizing your homie because he was dragged through the streets by a sadistic enemy, and in the next mission you can be spraying sewage on pedestrians. The vision and tone of that game was all over the place.

CLICK

- This stupid tech debt thing hurt us again.
 - We had to fix the streaming issue by making all the city chunks a uniform size, and also made them smaller so we could have 3 loaded at once instead of two, and then placed them in a brick pattern so you could never run across more than three at once... and needed to fix holes in the world to accommodate flying.

- But as with SR1, we did a poor job of actually quantifying this technical debt and work. Fixing these issues in a small section of the world – the Vertical Slice - doesn't do a good job of telling how much work there is when you scale up for then full world.

THE LESSONS

- Player Experience Matters
- Develop Deep. Not Wide
- Quantify Technical Debt
- Check Egos at the Door



IN HINDSIGHT...

Maybe this wasn't such a good idea after all.



The key lessons from the Saints Row 2 Vertical Slice:

You need to work towards delivering a player experience - Just having features in a vacuum doesn't cut it.
If you focus on the player experience, the necessary core gameplay and features and tech you really need will follow.

CLICK

We did the whole shotgun approach and developed wide, not deep
The methodology was: throw out ideas and see what sticks.

CLICK

We needed to improve at extrapolating the technical issues. You won't be feature complete at a Vertical Slice, but which critical systems are really still their early states? And how much work do they really need?

CLICK

And finally:

Work with previous teams. “Not Invented Here” is a detrimental attitude. Share learnings and mistakes. Competition between teams can be a good thing, but not if it means losing productivity and experience

FUN FACTS

- Co-Op
- 3DS Max
- 94,000 Reported Bugs



Fun Facts for SR2:

- Co-Op was added while in production. Or at least at the end of pre-production. It was our attempt to implement a Silver Bullet feature. It wasn't accounted for at all in early schedules, and certainly not the Vertical Slice. Guess what – that added some unaccounted work.

CLICK

- As with Saints Row 1, we used 3DS Max to create and edit the world.

CLICK

- SR2 reported 94000 bugs. The mean bug-count for all other Volition projects is below 40k bugs. Which begs the question – did we have a better QA department on SR2 or was it a buggier game?

Next up is red Faction Guerilla



Next Up: Red Faction Guerilla – I'm going to refer to it as RFG or Guerrialla
June 2009 – 360, PS3, PC

THE HISTORY

- More Hard Stuff
- Open World Destruction



RF2 was another challenging project for the studio.

A key hook in this game was destructible environments.

We went from Destruction in a linear game (RF1 and RF2) to Destruction in Open World – Which is a very hard problem to solve

Think about it – Real-time, physics-based, arbitrary destruction can be complicated and challenging enough in a linear game (RF1 and RF2).

But the studio was moving towards making open world games exclusively and it was time to take the destruction technology into that genre.

And it took a long time to figure that out and make it work.

THE PLAN

- Tech Demo
- Solve Open World Destruction



Because the problem of figuring out Destruction in an Open World was so challenging, the plan was, quite intuitively, to first get the destruction working in a technology demo

Lots of tech demos. The team went through at least four of them.

To be clear – these tech demos weren't Vertical Slices. These were demos focused on getting the destruction technology working in an open world environment.

On the next slide I'll show you a clip from one of these demos...

Well, to be fair, this clip is more from design than the programmers on the team.

Bear in mind: The design team was trying to figure out how open world destruction could actually work **while** the technology was still being figured out.

So you'll notice this video is less an actual gameplay demo and more a video that shows what the gameplay **COULD** be like, based

on where the technology was at the time.

And – it doesn't have any sound.



Before 26:00

THE GOOD

- Spent Time Getting it Right
- Solved Open World Destruction

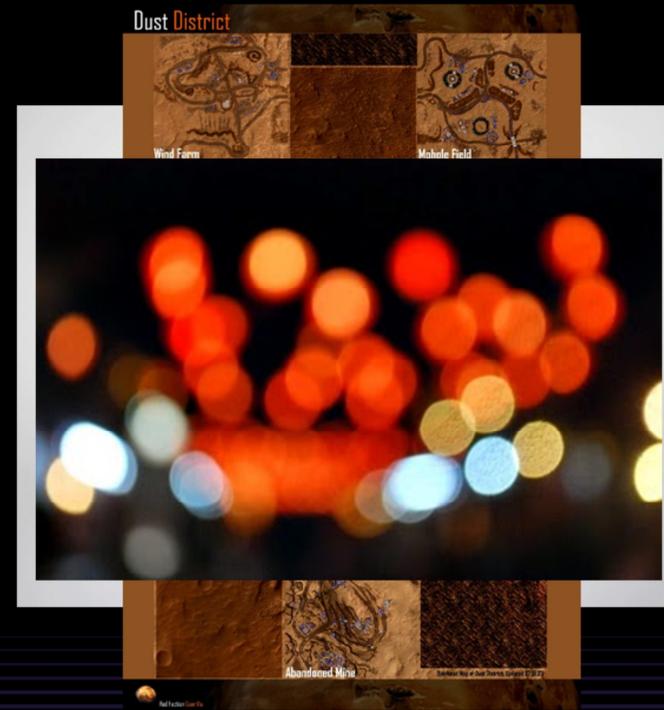


So, looking back at RFG and it's vertical slice hindsight:

- The team got the time to “do things right”. Which in this case was:
- Solving the very hard problem of open world, real time physics-based destruction.

THE BAD

- No Real Vertical Slice*
- Mushy Vision



As for what didn't work:

- There was no true vertical slice. Well not before production.
- After the technology was “completed”, the team went into production.
- But – there were still a TON of design issues to answer and explore yet.

CLICK

- This is a map of the RFG world section called Dust
- The team re-made this map at least 4 times in “production”
- BECAUSE they hadn't figured out the design, the gameplay yet.
- And effectively, the re-doing of Dust became the project's Vertical Slice, even though the team was officially in production.

CLICK

- There was a Soft player vision. Some on the team had a running joke:

- What is this game about? Well, You destruction!
- What do you do? You destruction!

THE LESSONS

- Open World Destruction is **HARD!**
- Tech Demos are not a Vertical Slice



HINDSIGHT

If he is going to install the button anyways why doesn't BOWSER step on it when Mario gets on the bridge?



So, the key take aways for RFG:

- Open World Games with Destruction are Really Hard and Take a Long time
- So long, that sometimes you'll focus on a singular problem – in this case the tech - and forget about the intended experience in preproduction.
- The tech demos allowed the team to solve the tech problem... BUT

CLICK

- Tech demos are not vertical slices precisely because they only focus one thing – solving technology problems. And while this can be an important piece of the pie, its only one piece.
- Basically: You can't prove out the game design when tech of this magnitude is still evolving and being figured out. And you can't say the team is in production if you're design is still being proven out.

FUN FACTS

- Two SR2 Games = One RF Game
- Originally a First Person game



Two Saints Row games came out before RFG did. RFG's development took a bit longer than 5 years. We like to tease the RFG team about this. They do not think it's as funny as we do.

CLICK

RFG was originally a FP game. And then the publisher asked to change it to 3rd person because at least on console, 3rd person games were selling better than first person. This mandate produced a short lived shit storm at Volition, but in retrospect, they were probably right.

Red Faction Armageddon is up next



June 2011
PS3, 360, PC

30:00

THE HISTORY

- Challenging Project
- Publisher 'Collaboration'



COLLABORATE

Sometimes your best option is to say:
"This is all her fault."



Depending on who you ask, some will say RFA was a more difficult project than SR1

There was a lot of "collaboration" from the publisher. Unfortunately, much of this collaboration came in the form of specific feature ideas.

THE PLAN

- Red Faction Crusade
- Vertical Slice:
 - 4 months
 - 25% above ground
 - 75% below ground
 - Visuals



Red Faction Armageddon was originally planned to be Red Faction Crusade – a direct sequel to Guerrilla. The producer was from a company that also did Vertical Slices as part of the development cycle. And by this time, vertical slices or 1st playables or whatever were part of our standard development schedule.

CLICK

The plan:

- 4 month deliverable.
- 25% above ground gameplay
- 75% below ground gameplay
- prove out underground mechanics (open destruction underground = what?)
- Some new single player features like “Repair” (in addition to breaking things with destruction you could also

build things up)

- make it better looking than RFG

On the next slide you're going to see a quick clip of footage from RF Crusade's Vertical Slice build. We're showing this because it helps solidify how far along this game was when it was working on the Vertical Slice. It looks good and lots of features and core gameplay are up and running. This particular clip is showing a section of underground combat with a focus on a new weapon called the Magnet Gun



Before 31:00

THE PLAN.. WRENCHED



“One word: Aliens...”



Team received some creative notes from a newly hired Creative Director at THQ. These notes came during month 2 of Red Faction Crusade's 4 month Vertical Slice.

CLICK

One of the big motifs from these notes was: Get some Aliens in the game.

At the time, the team was thinking something like: Okay, thanks for the notes, but surely these can't be a mandate. We're two months away from production and in the middle of a vertical slice.

So the team kind of ignored these notes things kind of drifted away for a couple of weeks.

-Shortly thereafter, THQ went through some management changes and that Creative Director became an EVP

-And then even more shortly thereafter the team received some more creative notes. And they were: Did you think I was joking about aliens?

CLICK

-Epiphany! Ah... we get it. We need to add in aliens.

And there was also a rather important footnote to the whole "add Aliens thing". That was:

You need to bring the game in by six months...

THE NEW PLAN



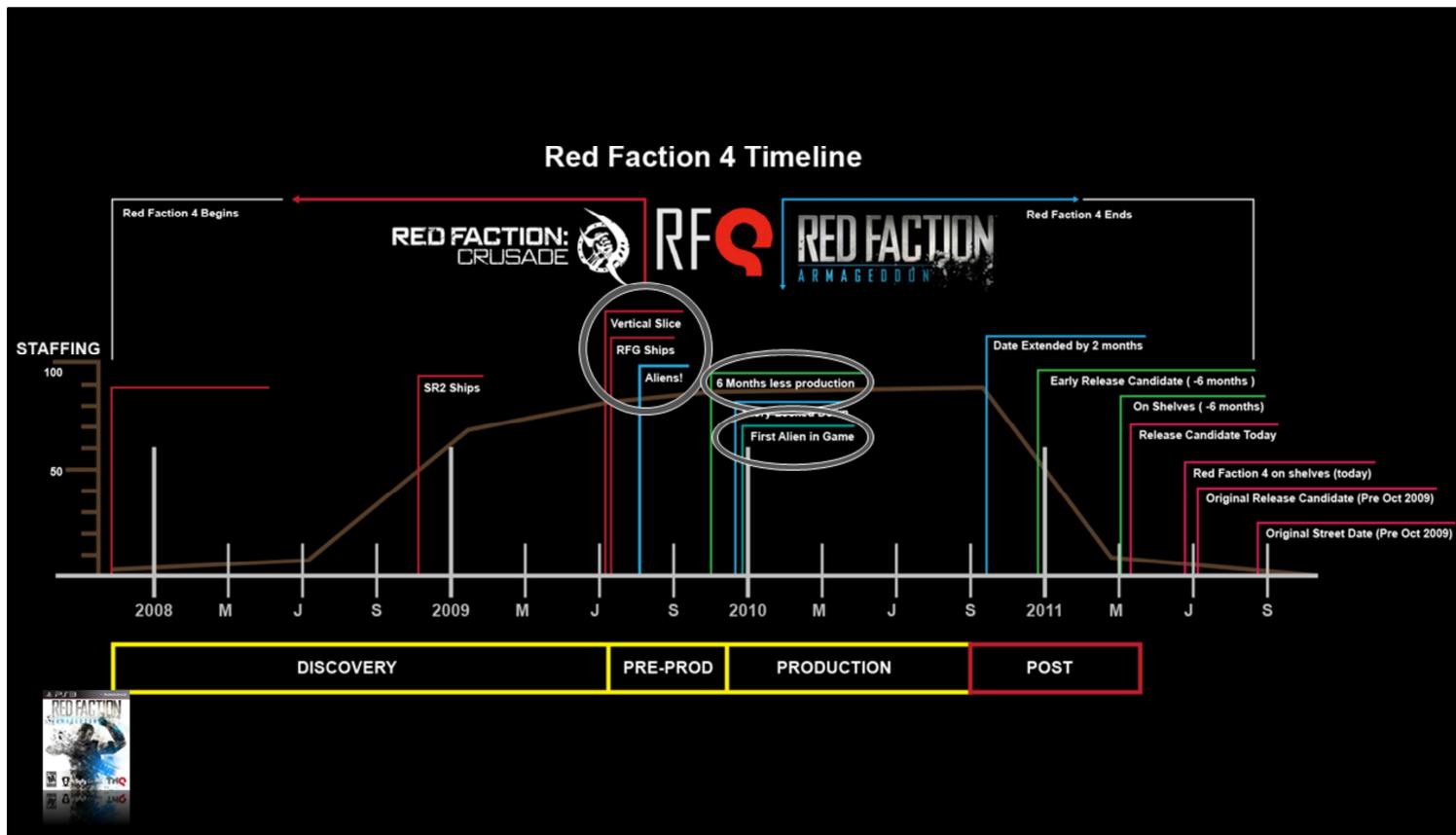
UNDERGROUND



So. As you might imagine, there was some scrambling.

And the team and Studio Management came up with a New Plan. And that was:

- Solution (out of necessity):
 - Make the game linear
 - Take it underground (because linear would be easier underground) and
 - Get the Aliens in game



Here's a way to contextualize what happened:

- Here's where the Red Faction Crusade team was working on their vertical slice (month 2 of 4) when the request for Aliens comes in.
- And here's where, for various reasons, the team needed to pull in their street date by six months.
- And here is when the team got their first Alien in the game

THE GOOD

- Team got in aliens
- Team brought ship in by 6 months (!!!)



What worked:

The RFA team got in ALIENS!

CLICK

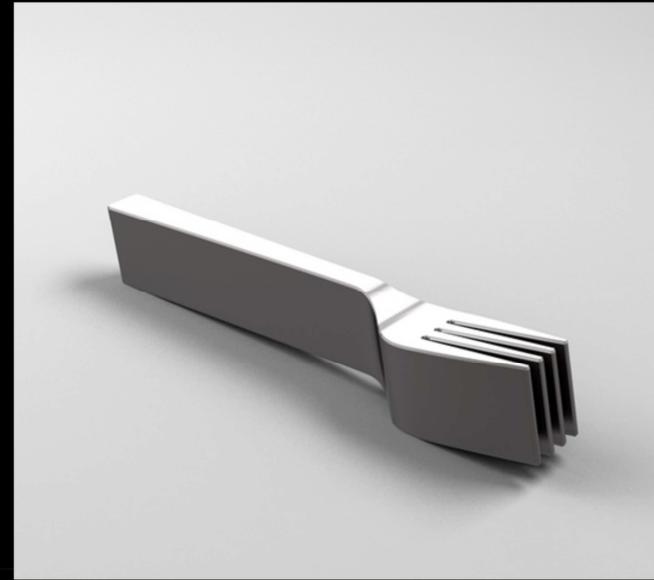
And pulled in the ship date by multiple months

Made the necessary but painful decision to go from open world to linear in order to hit a date

Seriously – what this team did was pretty incredible. I personally have never worked on a game of this size that brought in the street date by six months and added in a crazy new feature like Aliens at the start of production.

THE BAD

- No Vertical Slice



What didn't:

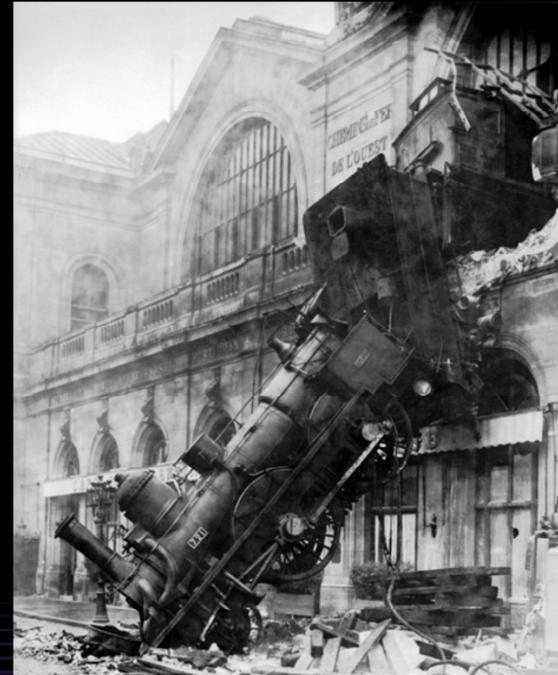
There was no Vertical Slice. This was the epitome of a team figuring things out as they went. Aliens were added in production. They abandoned their plans to do a vertical slice and just plowed ahead.

Right or wrong – that was the decision made then. And because of this, it wasn't an easy production schedule.

And when you're figuring out things like "What do the Aliens look like, how do they move, what are they?" you are most certainly not in production.

THE LESSONS

- Shit Storms Happen
- “Production”



The Key lessons on RFG:

Sometimes despite best efforts and good planning, the shit can hit the fan and everything falls apart

You can say you're in production, but if you're figuring out pretty much all the things as you go, you're not in production and are fooling yourself.

To be clear: It was well within the publisher's purview to mandate Aliens. And to be fair – the publisher did not dictate the behavior or design or art direction of the Aliens.

And Aliens aren't exactly out of place in a Science Fiction game.

The issue was the request came too late in the team's development cycle.

FUN FACTS



Okay, this really isn't a fact, but it's fun.

Once the request for Aliens came in, this clip made the rounds at the studio very quickly.

Would you call this serendipity? Ironic?

Okay, now we're at SRTT

Before 35:00



SRTT
Released in November 2011
PS3, 360, PC

THE HISTORY

- Same Leads as SR2
- Lessons



SRTT:

- Largely SRTT had same leads team as SR2. This was helpful in that:
 - We were able to more directly learn from mistakes on SR2 development
 - Experience told us we needed a more defined Vertical Slice
 - Especially: focus Aesthetics (SR2 is not the prettiest game; desired to be more competitive)
 - Started planning for multi-phased Vertical Slice

THE PLAN

- Multiple Vertical Slices



COMPETITION

Between good friends...



The Plan was, once again, to deliver the vertical slices in phases.

But this time we were going to do a two prong attack: focus on art and gameplay separately, and then bring them together.

The thinking was: let the artists focus on making things look good, and let design focus on the fun factor. Let's minimize interruptions, do our respective things, and then marry them as a final POC deliverable.

On the next slide is an example of one of the Art Vertical Slices. This was an earlier one, designed to show off improvements to visual acuity, particularly within environments.



This next slide is an early part of the Design Vertical Slices

On SRTT we tried to focus more on the gameloop (what the player is doing over and over again in the game) at least when compared again previous SR games.

This example you'll see shows when we were working on transitions in the gameloop – the player needs to go somewhere and then there is a cinematic at the destination – how would that feel and what would the transitions look like?

This clip is one of many of the design Vertical Slice pieces the team made on this project



Earlier than 39:00

THE GOOD

- Focus on Visuals
- Art and Gameplay Slices
- Gameloop



What worked:

- Definitely showed improvements in the visual quality, at least when compared to Saints Row and Saints Row 2

CLICK

- The separate art and gameplay vertical slice approach was beneficial in that each discipline was able to focus on their respective areas without worrying about one another (at least for a time). It was a helpful approach.

CLICK

- SRTT marked the first time (gameplay side) – at least in the SR series - we focused more on the main gameloop. And for SRTT, that gameloop consisted of:
 - Customizing your character
 - Going to an objective
 - Watching a cinematic

- Killing things
- Getting stuff

And because of this focus we ended up focusing more on a holistic Player Experience. It's a subtle thing, but even transitions – as shown in that clip – do contribute to the overall player experience.

THE BAD

- Performance Budgets
- Smoke and Mirrors



What Didn't

- We didn't spend enough time nailing down budgets. Not monetary budgets – performance budgets. Those interiors looked great. But at the end of the day: this was a mistake because they weren't realistic. We shouldn't have cared about individual assets and instead focused on the overall scene.

Instead of making the bananas look super-awesome, we should have spent more time up front focusing on things like:

- LODs
- Draw distance
- Ambient spawning peds and vehicles
- To have painted a more realistic scene. Instead we created an environment that was unrealistic.

CLICK

- So, and at least for the art-side, instead of creating a vertical slice we created an aspirational visual bar. A visual bar is not the

same thing as a Vertical Slice. Visual bars are not a bad thing, but they are a different thing. We had smoke and mirrors in that our art Vertical Slice wasn't representative of final quality precisely because it wasn't a realistic depiction of what we could show in the final game because we didn't spend enough time of performance budgets.

THE LESSONS

- Performance Budgets Matter
- Visual Bar is not a Vertical Slice
- Game Loop



HINDSIGHT

Those really were the droids you were looking for.



Key lessons for SRTT's Vertical Slices:

- Again, and for SRTT, a specific category was performance budgets. We didn't do a good job establishing realistic targets. Many on the team, as with SR1, spent a large amount of time optimizing performance in production. Granted that work this time around wasn't nearly as brutal (because we were more experienced with these systems), but it was nonetheless work that could have likely been mitigated had we done a better job at analyzing how expensive our scenes were in the vertical slice.

CLICK

- Related – establishing a visual bar (which is effectively what we did on the art side) is not a vertical slice because your bar may not be possible or realistic.

CLICK

- Finally – working on the main Game Loop was helpful. In addition to necessarily requiring the implementation of key features

to complete the Game Loop, it also forced us to think about commonly overlooked things like “navigation duration” and the transitions from gameplay to cinematics – all of which contribute to the overall player experience.

FUN FACTS

- Fire Retardant Burt



Spoiler Alert! - Burt Reynolds is a character in Saints Row the Third

Contractually his character a) couldn't use firearms (he could only punch) and b) he couldn't catch on fire. He could die, but he needed to be impervious to fire.

Last, but not Least...

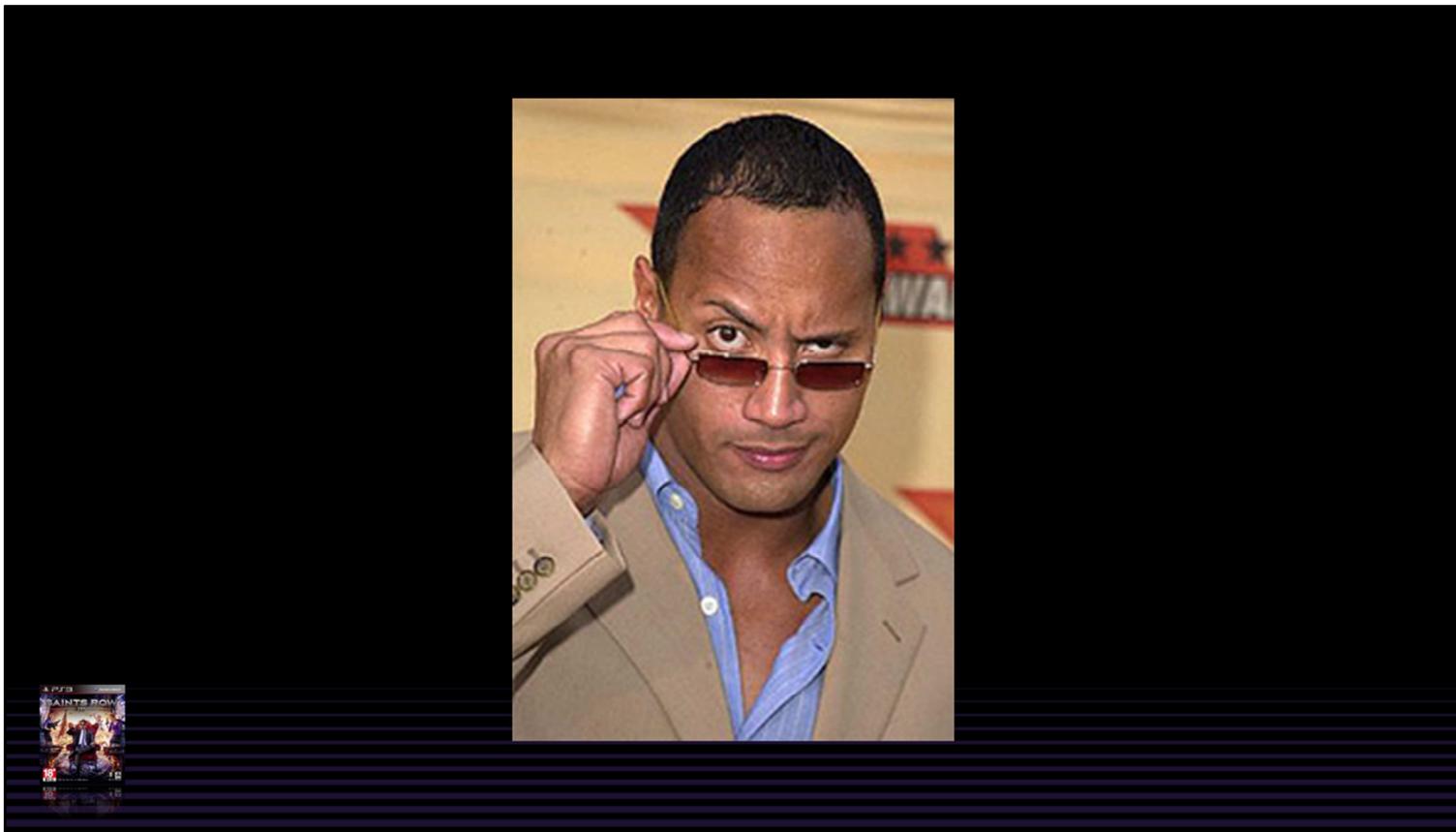


SR4 – August 2013
360, PS3, PC

THE HISTORY



SRIV got the best, most impressive and thorough Vertical Slice DSV has ever made.



Oh really. How did that happen?

(Glad you asked... otherwise I would have been missing a segue here)

THE HISTORY



(I covered some of this last year, but it's so wonderfully amazing it bears repeating)

The SRTT team started on a stand-alone expansion pack after SRTT shipped, called Enter the Dominatrix

ETD was a) in the same city as SRTT, b) and limited to new tech/features.

The stand out features were Super Powers and Aliens. The core of the game was, in fact, "The Saints in the Matrix. With Aliens"

The super powers were fun. The title was shaping up.

And then for business reasons, THQ decided to take what was best about ETD and add it to SR4

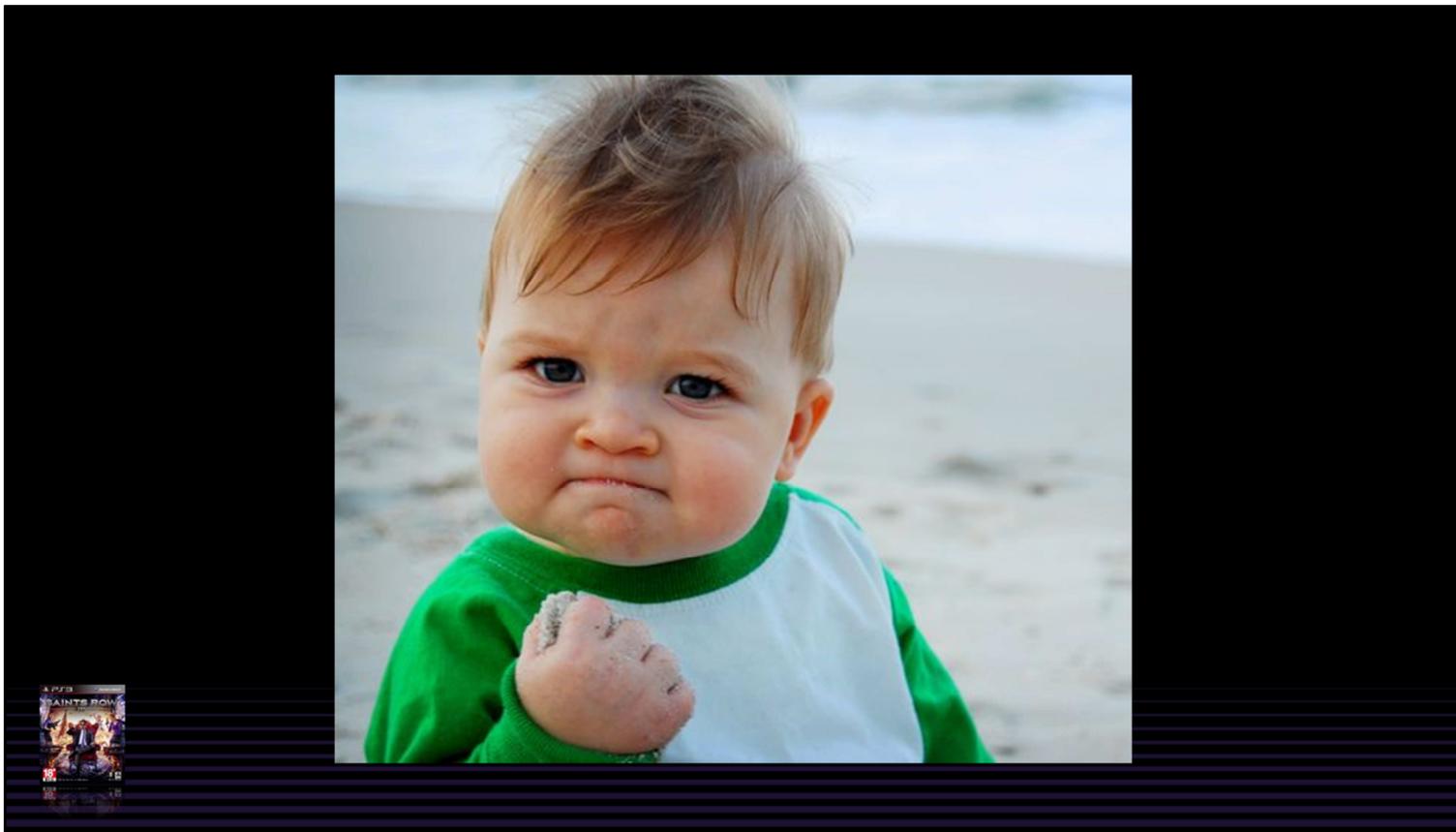
However, SR4 was originally doing it's own thing. It was a separate and large team going in a direction without super powers.

So effectively the original SR4 team got word they needed to take **some** of what they had and combine it with much of ETD. But they didn't get extra time to hit their street date. Basically they had 1 year from the time the decision to combine the two

was made.

That's not a lot of time to be adding something like... SUPER POWERS and ALIENS in your game when you weren't planning on it.

But on a positive note, SRETD was pretty far along and could be used as a rather marvelously fleshed out Vertical Slice for the SR4 team



Unbelievable. Think about it – at least from a Vertical Slice point of view.

Imagine that your vertical slice is well over 100 man-months of focused work by experienced people who fully understood what they were working on.

So, in this vertical slice you got:

- A playable critical path playable (which amounted to about seven hours of gameplay),
- More than functional new features, (i.e. the Super Powers)
- An art directed city.
- And a preliminary quantification on performance budgets.

Greatest. Vertical. Slice. Ever.

THE GOOD

- Vertical Slice Level 5000



So, looking back at things on SR4. The good was:

- It was the greatest VS in the history of this industry. The SR4 team had nearly everything from a Vertical Slice point of view laid out and handed to them.

THE BAD

- Anomaly
- Financially Untenable



What didn't work:

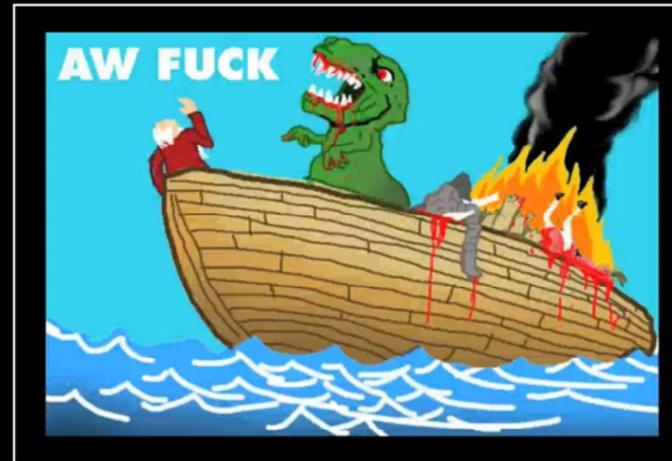
- This kind of VS doesn't work unless you have near limitless time and money. And unless you work at a studio whose name rhymes with... Malve, you probably don't have near limitless time or money.

CLICK

- Think about it – that more than 100 man-months on SRETD wasn't just time, it cost money too. Depending on your studio burn rate, it could be a lot of money. And that makes it financially untenable. At least for most of us.

THE LESSONS

- Ideal Model
- Unrealistic Model



The biggest take away was that SR ETD definitively showed how useful a properly constructed VS could be.

We had:

- Player experience demonstrated through a fully playable critical path
- Core Gameplay proven and iterated with some polish
- 7 hours of gameplay (demonstrated pacing, variety – lots of “things” that can be challenging to achieve in a VS deliverable)

CLICK

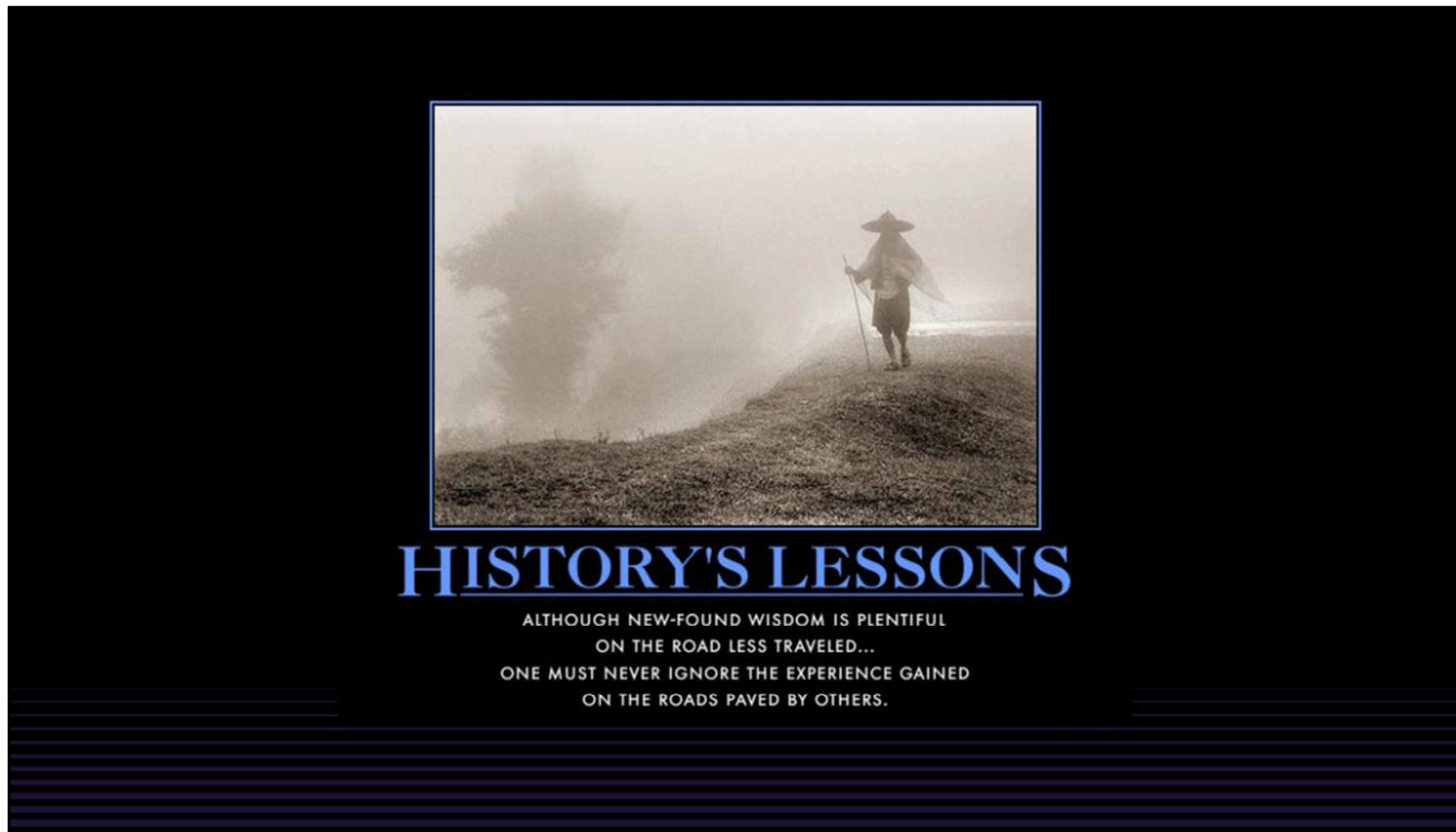
- BUT: This was nutty and we couldn't and can't expect to have Vertical Slices like this in our future.

FUN FACTS

- SRIV Classic



Before the addition of super powers, the original design in SRIV (internally we now call it SRIV Classic) called for warring tribes that were headed up by historical figures. These figures were cloned and brought back to life by Ultor (a shell corporation for the bad guys in SR) and included the likes of: Genghis Khan, Hitler, and Cleopatra.



Volition is currently a one project shop. These days, we have a larger collection of experiences now that we've made more games, and one goal – one thing of many Volition is pretty good at - is to better learn from the past and so we can improve and have best-practices on future games.

This philosophy applies to many areas, and the Vertical Slice is not exempt.

What key Lessons have we learned? And what are we doing in regards to a Vertical Slice milestone today based on our past mistakes, trials and tribulations?

*Again, this is what works for us and the types of games we make. YMMV

JUST DO IT
(TRY TO) DO IT RIGHT
(SOME) FAILURE IS INEVITABLE

46:00 is good

These are the basic, and top-level lessons or best practices we believe in based on our past experiences

And I'll go over a bit more in detail in these next slides.

I tried making a grid or matrix that showed what learnings happened on which project, but it just didn't make sense since the experiences, lessons-learned and best practices

In our history is, frankly, all over the place.

JUST DO IT

- The Vertical Slice is important
- Plan for it
- Make it a Visible Goal
- Trying counts



TEAMWORK

Teamwork Means Never Having To Take The Blame Yourself

Apologies to a certain shoe company

The Vertical Slice Milestone is important.

Plan for a it.

Make it a real thing.

Make it a part of the team's schedule and make sure the team knows about it.

We believe and have found - There are too many benefits to NOT do it

If you think about it, a vertical slice deliverable will happen regardless at some point in the project.

If you PLAN for it and attempt to nail it before production, you will have likely saved the team some pain down the road

We have never done it as well as written on paper, BUT not trying would have been worse. Just talk to some Red Faction Armageddon team members.

Even with our issues and failures, we still uncovered issues and problems we otherwise would have uncovered later in production. When we haven't done a vertical slice or attempted one at all, things have not gone well.

(TRY TO) DO IT RIGHT

- Production Gate
- Player Experience Focus
 - PX -> Core -> Features -> Systems
- Quantify Tech Debt
- Longer than you Think



Plan on doing it “right*”.

This means

- Use it as a Gate to Production
- If the Vertical Slice doesn't help your team:
 - Understand what they are making, How they are making it , and With an Acceptable level of tech debt
- Then you're not ready for production

CLICK

To reiterate: Focus on a Player Experience. Not a list of features.

If you define the Player Experience first, the features, core gameplay, and tech needed to support these will necessarily follow

CLICK

Quantify your existing technical debt. Doing this well is a new and hard thing for us. Again – no way does the Vertical Slice Milestone mean you are feature complete. There will be systems and feature and tools work in production. But you should work

with the team to determine how you plan to quantify how much “tech work” you will need to absorb in production. Some is inevitable. How much will be dependent on your specific situation. And remember: Tech Debt here refers to establishing performance budgets in addition to correctly architecting systems and features.

CLICK

This deliverable will take longer than you think.

Make the vertical slice multiple milestones.

A good way to break it down for phases so the team can check in and shift priorities as needed: Get it working, Get it Working Right, and then Polish it

(SOME) FAILURE IS INEVITABLE

- Too Big
- Too Many Questions
- Too Many Things



PLANNING

MUCH WORK REMAINS TO BE DONE BEFORE WE CAN ANNOUNCE
OUR TOTAL FAILURE TO MAKE ANY PROGRESS.

This deliverable is too big and too complex and too early in development to bat 1000
You're going to miss some things
And you're going to fail in some areas
That's okay
We've never nailed this thing and yet still attempt it every time

Go into this thinking about the spirit of the deliverable:
Can you honestly say this team is ready to go into production?
Do the team have a solid idea of what it's going to take to make this game?
Does the team believe the Player Experience has been delivered well enough to move into production?

Doing a lawyerly checklist against specific assets in the game **as the only means** for evaluating your progress here will be futile.
You're too early in development for such things and the bigger picture of the deliverable matters just as much.

FINAL THOUGHTS

- Balance
- Goal:
 - Player Experience
 - Tech Debt
- Objectivity
- Worth it
- Still Improving



PERSEVERANCE

THE COURAGE TO IGNORE THE OBVIOUS WISDOM OF TURNING BACK.

Much of executing on the Vertical Slice deliverable comes down to balance.

Vertical Slice doesn't mean feature or content complete. It's not a finished game.

The key with this deliverable is to determine the most important things that need to be worked on in order to make production (the act of knowing what you are making and – importantly – how to make it) as painless as possible.

CLICK

Work with the team to help determine the End Goal for this deliverable, which in our opinion is:

- Realizing the Player Experience (which again, can't happen without necessarily defining your core gameplay, core loop, systems and features) through assets and gameplay that are representational of final quality. Make sure you aren't doing this through total smoke and mirrors. You need to have something that is realistic and not an impossible and unreachable bar.
- Quantify and track the team's Technical Debt. How much is acceptable? How much tech work will be done in production? Work with smart people to prioritize the systems and features you need furthest along (which in most cases will be the riskiest and most unproven)

CLICK

When you're done with the deliverable: Be objective. Analyze the results and have uncomfortable conversations. It will be worth it in the long run. If you rationalize where you are, you will set the team up for pain. Really quantify where you are after this deliverable. Do you need more time before going into production? If so, it's probably a good idea to get it if you can.

CLICK

Despite knowing you won't nail everything, it's still worth it. The scary things lying there waiting to be uncovered will be there regardless and you'll find them eventually.

The Vertical Slice can help find them earlier.

CLICK

Internally at Volition, we are still improving at how we tackle this beast. And despite all the angst and uncertainty that can come with it, we still attempt them.

WAIT! - MORE FINAL THOUGHTS

- Making Games is Hard
- It's worth it
- Vertical Slice can help



I personally believe: Nothing worthwhile is easy and there is no good progress without at least some struggle

I think the majority of us do this – making games - and get involved with trying to “schedule profitable fun” because: we love the end result of what we are creating, we like working with creative people (despite the inherent frustration that can come with that), and we love games.

It's too challenging otherwise. I mean, why sign up for this?

The Vertical Slice Milestone can help ease some of the pain of making games.

Hopefully you got something out of this talk – we're getting better at sharing experiences and best-practices (like through GDC), and that should mean better games, which is something I think everyone wants.

Thank-you very much for coming.

SPECIAL THANKS

- Conference Associates
- Volition – Past and Present

PLEASE FILL OUT COMMENTS!

Questions?

Take some o

Greg.Donovan@DSVolition.com



volition

