PART 5

Introduction

Front-end vs Back-end Developer Teams

Technical Methodologies

Quick revision

Quick revision

Besides visual appeal, what’s the purpose of animation on the web?

Quick revision

Besides visual appeal, what's the purpose of animation on the web?

- Communicate changes in state

Quick revision

Besides visual appeal, what's the purpose of animation on the web?

- Communicate changes in state

What are the drawbacks of animation?

Quick revision

Besides visual appeal, what's the purpose of animation on the web?

- Communicate changes in state

What are the drawbacks of animation?

- Variable performance, especially on mobile

Quick revision

Besides visual appeal, what's the purpose of animation on the web?

- Communicate changes in state

What are the drawbacks of animation?

- Variable performance, especially on mobile
- Can be distracting and slow down experience

Quick revision

Quick revision

What can / can’t be animated?

Quick revision

What can / can’t be animated?

- Position, opacity, color **can**

Quick revision

What can / can’t be animated?

- Position, opacity, color **can**
- Automatic height **cannot**

Quick revision

What can / can’t be animated?

- Position, opacity, color **can**
- Automatic height **cannot**
- Display block / inline **cannot**

Quick revision

Quick revision

Why do we use timeline animation?

Quick revision

Why do we use timeline animation?

- Enables complex story boarding

Quick revision

Why do we use timeline animation?

- Enables complex story boarding
- Can be paused / resumed

Quick revision

Why do we use timeline animation?

- Enables complex story boarding
- Can be paused / resumed
- Can be looped

A brief history of web dev

A brief history of web dev

- Until the commercialisation of the web in the mid-late 1990s the web was static and there were few, if any, specialist developers

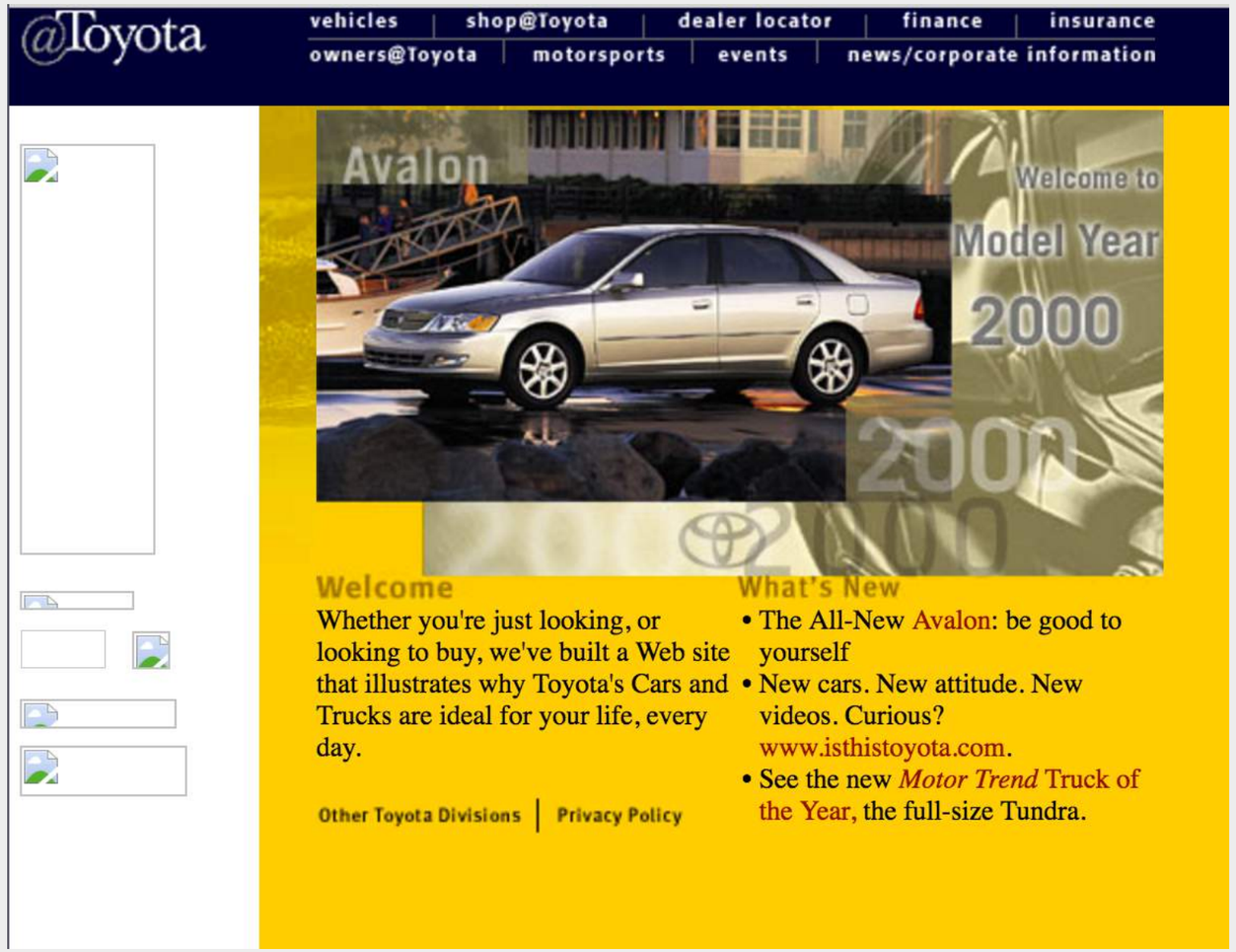
A brief history of web dev

- Until the commercialisation of the web in the mid-late 1990s the web was static and there were few, if any, specialist developers
- Until the late 1990s even the biggest companies only had a basic web presence

A brief history of web dev

- Until the commercialisation of the web in the mid-late 1990s the web was static and there were few, if any, specialist developers
- Until the late 1990s even the biggest companies only had a basic web presence
- The dot-com bubble led to a huge increase in demand for web development

Apple, 1997



Toyota, 2000

A brief history of web dev

The maturity of the web following the dot-com bust led to the evolution of web development practices including the specialisation of front-end and back-end developers.

PART 5

Introduction

Front-end vs Back-end

Developer Teams

Technical Methodologies

In a nutshell

In a nutshell

- Front-end development is concerned with everything that happens in the browser

In a nutshell

- Front-end development is concerned with everything that happens in the browser
- Back-end development is concerned with everything that happens on the web server

In a nutshell

- Front-end development is concerned with everything that happens in the browser
- Back-end development is concerned with everything that happens on the web server

Demo time

Back-end summarised

- Server, application, database

Back-end summarised

- Server, application, database
- For each request to the server the application will access the database, retrieve any content and then this will be processed and sent to the browser

It’s complicated

It’s complicated

- The introduction of Node.JS means we use Javascript on the server (back-end)

It’s complicated

- The introduction of Node.JS means we use Javascript on the server (back-end)
- This has also led to a concept called ‘universal’ web applications where front-end and back-end code is shared

It’s complicated

- The introduction of Node.JS means we use Javascript on the server (back-end)
- This has also led to a concept called ‘universal’ web applications where front-end and back-end code is shared
- This has led to some merging of the roles

PART 5

Introduction
Front-end vs Back-end
Developer Teams
Technical Methodologies

Wtf is Agile?

Wtf is Agile?

- A set of principles for modern software dev

Wtf is Agile?

- A set of principles for modern software dev
- Agile is meant to be what it sounds like;
Lightweight, fast, flexible, responsive

Wtf is Agile?

- A set of principles for modern software dev
- Agile is meant to be what it sounds like;
  Lightweight, fast, flexible, responsive
- Implemented in response to the connected
  and frequent nature of software delivery

Principles of agile

Twelve key principles including:

Principles of agile

Twelve key principles including:

- Customer satisfaction by continuous delivery and improvement

Principles of agile

Twelve key principles including:

- Customer satisfaction by continuous delivery and improvement
- Close co-operation of design, dev and biz

Principles of agile

Twelve key principles including:

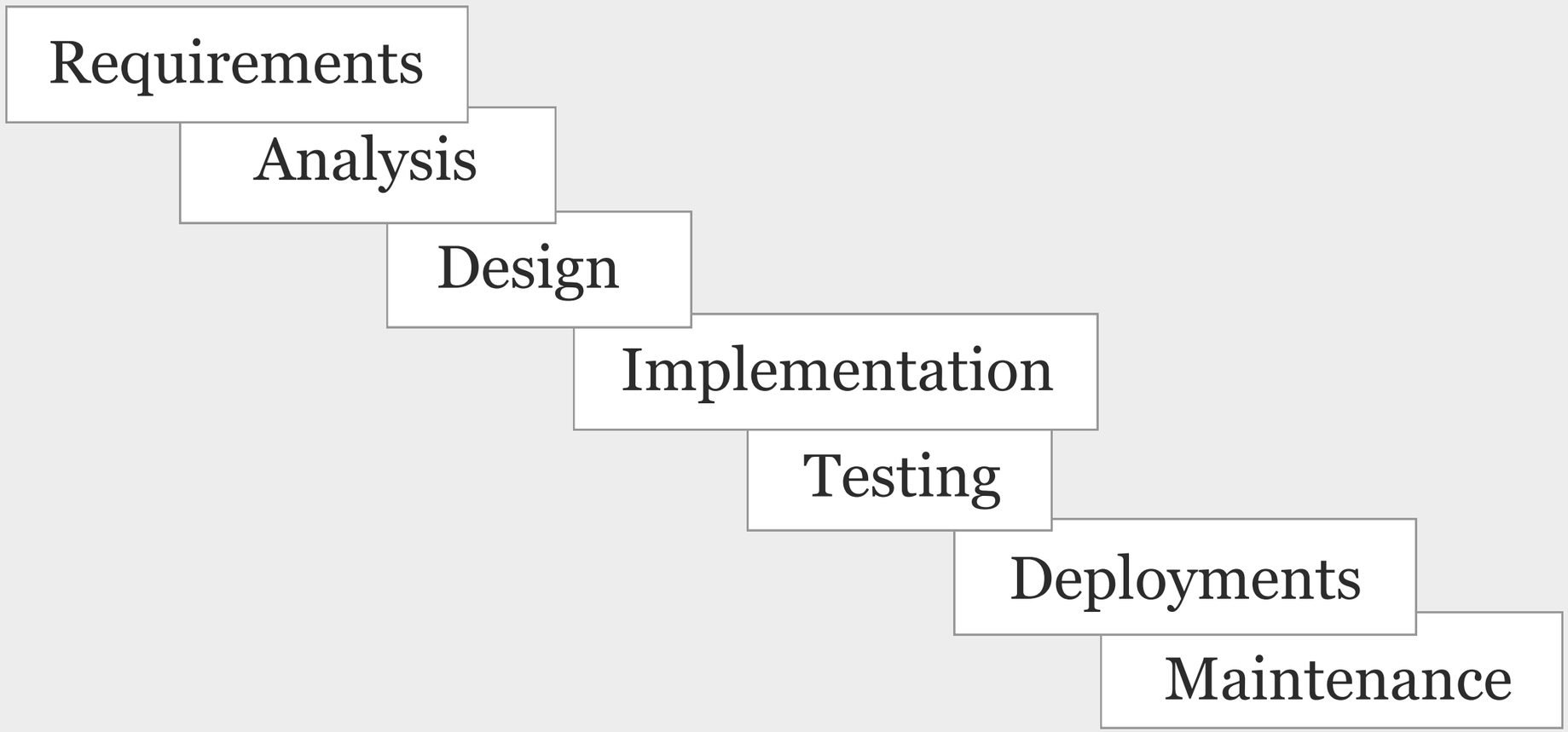
- Customer satisfaction by continuous delivery and improvement
- Close co-operation of design, dev and biz
- Changing requirements are welcome, even late in the project

Agile vs Waterfall

Agile vs Waterfall

Waterfall is an opposing approach that follows a sequential approach and doesn't allow for change or flexibility. If something changes then the process must revert to the start.

Waterfall processes



Agile rituals

Agile rituals

Supporting the principles of agile, the following practices are applied by Nine Digital

Agile rituals

Supporting the principles of agile, the following practices are applied by Nine Digital

- Daily ‘stand-up’ to allow all team members to understand current progress

Agile rituals

Supporting the principles of agile, the following practices are applied by Nine Digital

- Daily ‘stand-up’ to allow all team members to understand current progress
- Two-week ‘sprints’ to complete ‘stories’

Agile rituals

Supporting the principles of agile, the following practices are applied by Nine Digital

- Daily ‘stand-up’ to allow all team members to understand current progress
- Two-week ‘sprints’ to complete ‘stories’
- Retros at the end of ‘sprints’ to get feedback
- MVP principle

Agile planning and workflow

- Features / requirements are divided into ‘stories’ that are assigned a value to indicate the likely cost of developing that feature
- ‘Stories’ comprise two-week sprints and are assigned to a single developer to own

Which leads to...

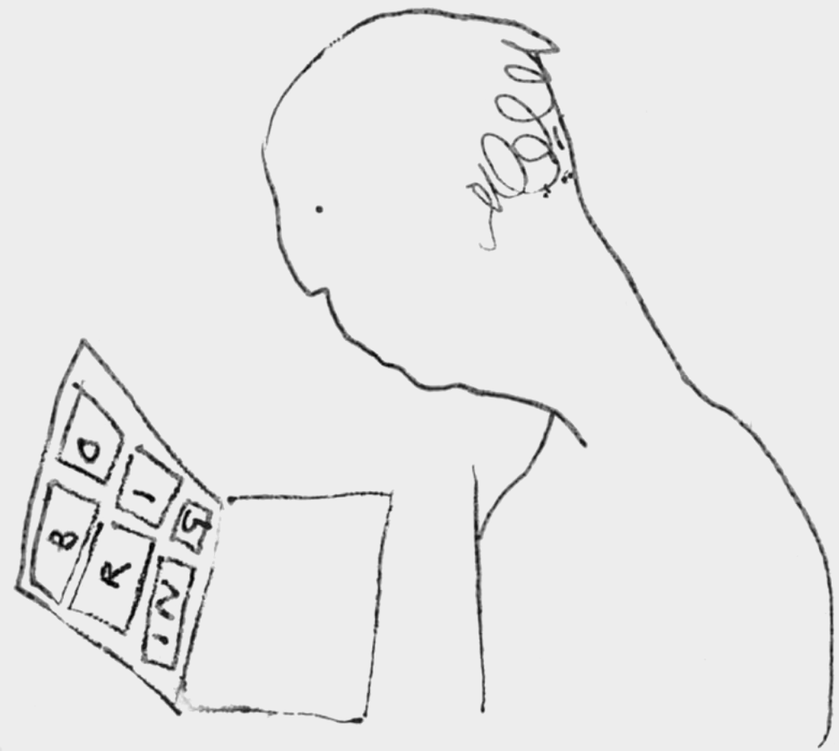
JIRA...

JIRA...

- Extremely boring software

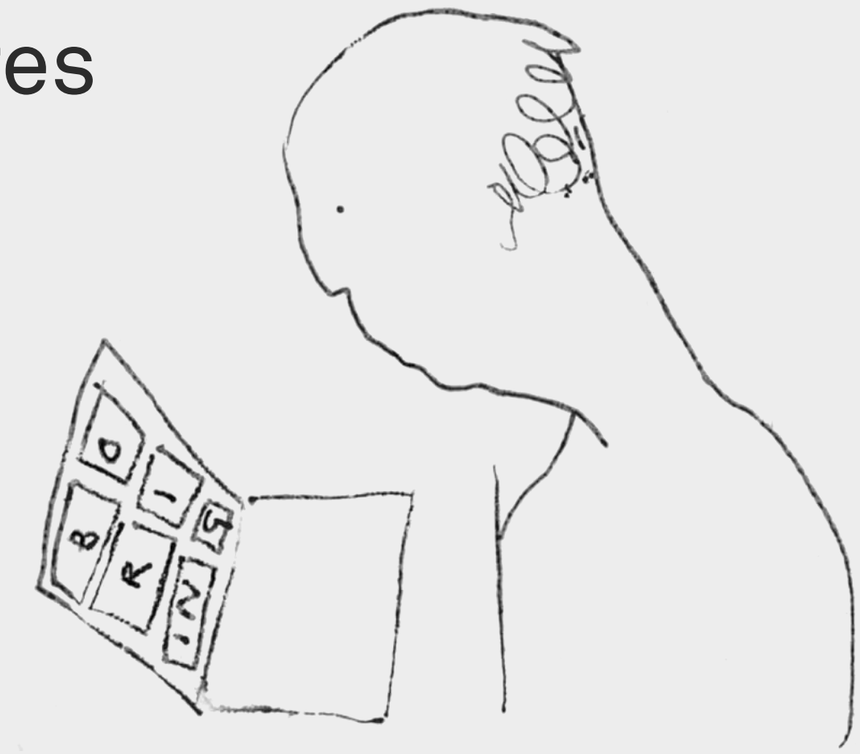
JIRA...

- Extremely boring software



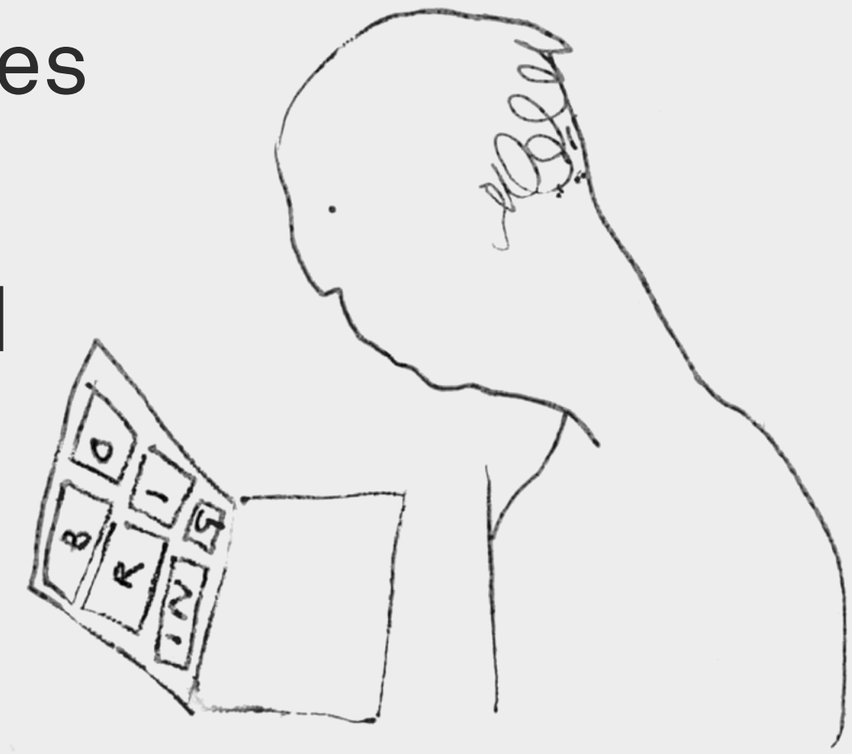
JIRA...

- Extremely boring software
- Tracks all issues / features and assigns developers



JIRA...

- Extremely boring software
- Tracks all issues / features and assigns developers
- All features are assigned using an agile-ish user centred approach



PART 5

Introduction
Front-end vs Back-end
Developer Teams
Technical Methodologies

What is Git?

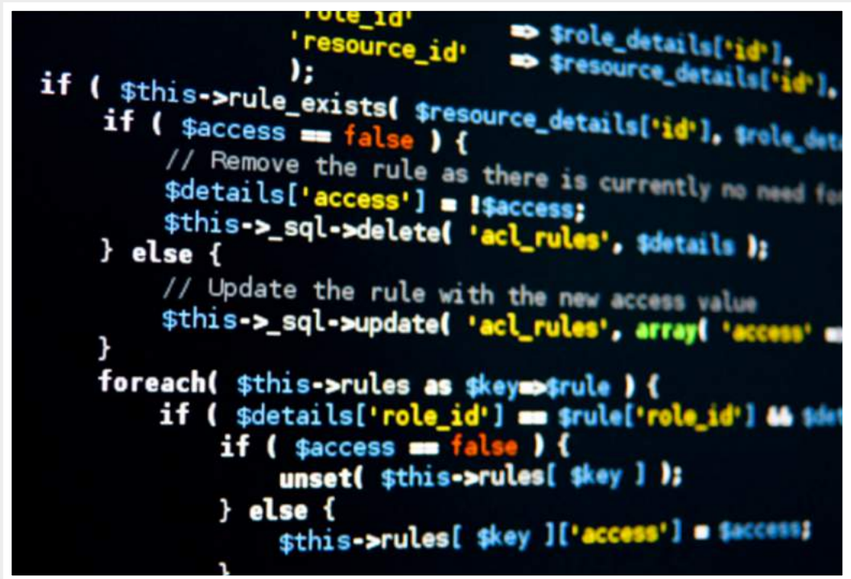
What is Git?

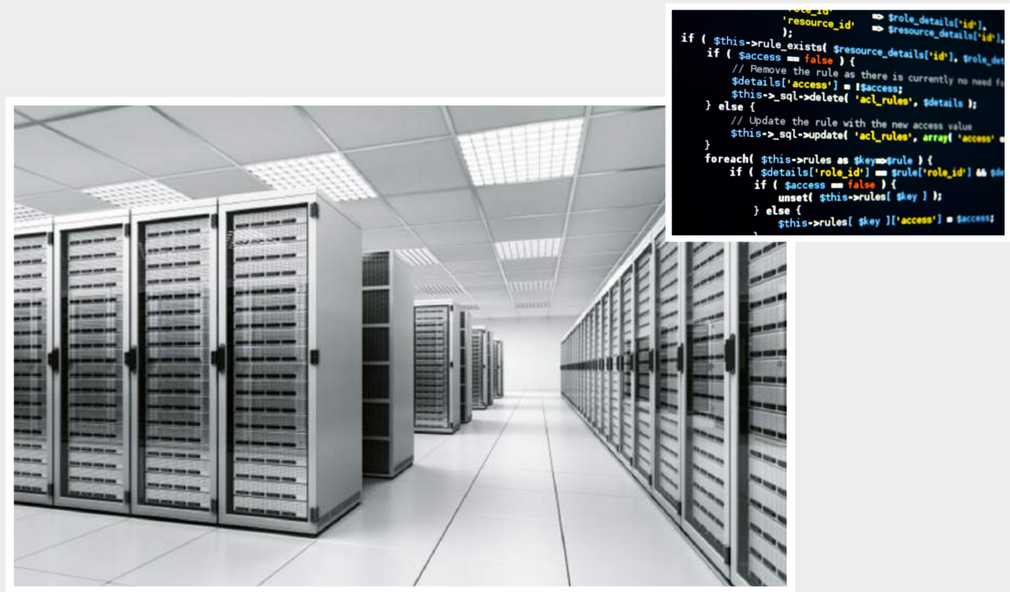
- Git is our version control system

What is Git?

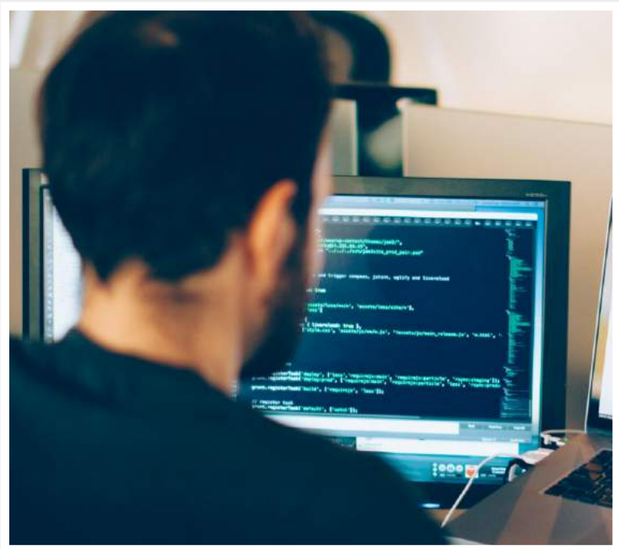
- Git is our version control system
- Version control is like....

Starting with an application's code...

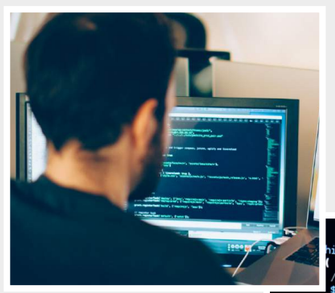


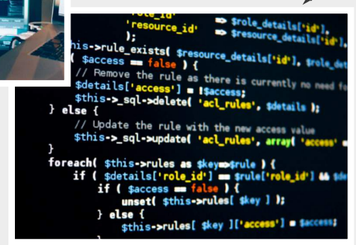


The application code is stored in a Git repository



Then consider a developer who
is contributing to a project

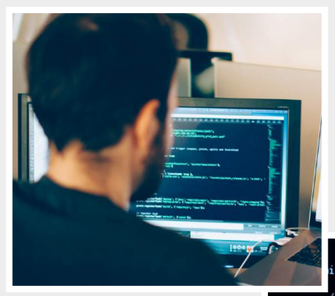


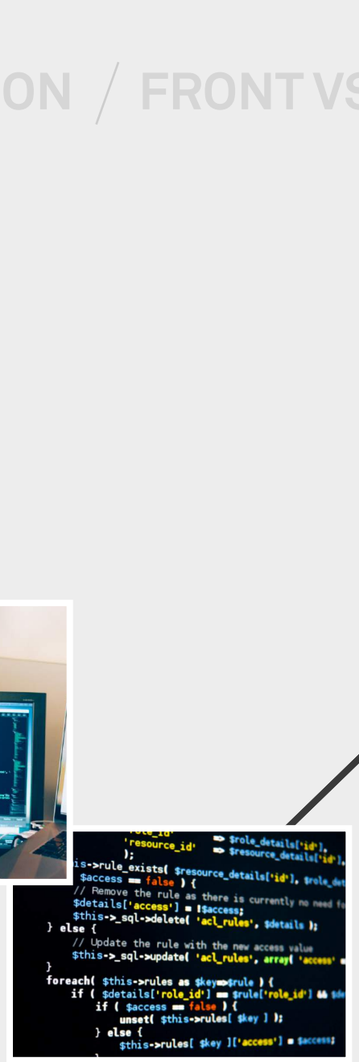


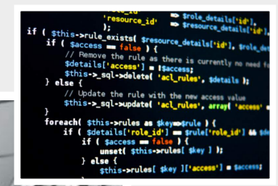
The dev ‘checks-out’ a copy of the code from the Git repository

The dev creates a ‘branch’ from ‘master’ and works on this locally

Once the dev is ready they'll submit a 'pull request' to other devs.







Once the ‘pull request’ is approved the code will be merged into the ‘master’ branch

Source control

Source control

- Every commit (code change) by a dev is logged

Source control

- Every commit (code change) by a dev is logged
- Pull-request reviews ensure the code on master branch maintains its integrity

Source control

- Every commit (code change) by a dev is logged
- Pull-request reviews ensure the code on master branch maintains its integrity
- Any changes that cause bugs can be identified and isolated

Source control

- Every commit (code change) by a dev is logged
- Pull-request reviews ensure the code on master branch maintains its integrity
- Any changes that cause bugs can be identified and isolated
- Any breaking changes can be wound back as the code can be reverted to any stage

Source control

Demo time

Thank you.