COMP30670
Software Engineering (Conversion)

12 - Project Description
02/03/2017

Dr. Aonghus Lawlor

aonghus.lawlor@insight-centre.org

Project

School of Computer Science
COMP30670: Software Engineering (Conversion)
Semester II

Assignment 4 (Project)
On: 02/03/2017
Due: 23/04/2017

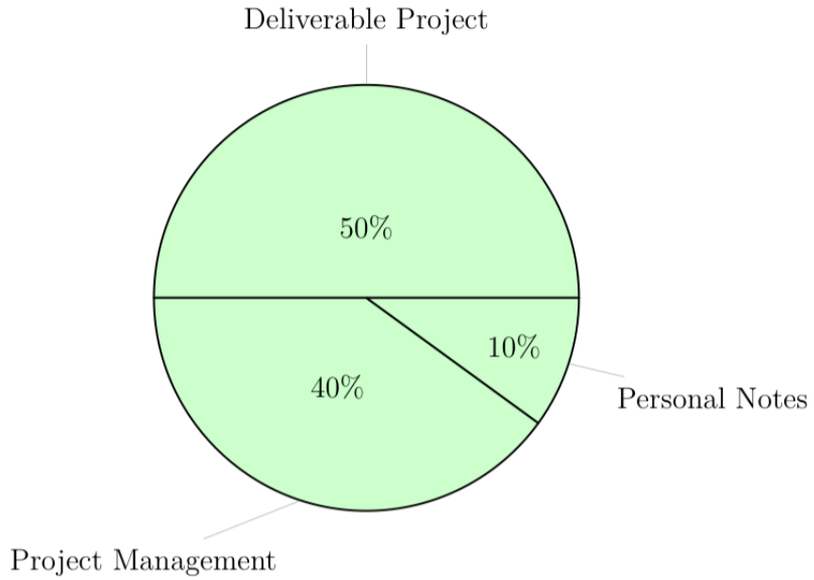
Q1: _____ (**100points**)

This assignment describes the project work for COMP30670. Refer also to the lecture notes on CSMoodle.

The goal of this project is to develop a web application to display occupancy and weather information for Dublin Bikes.

You will conduct the project in a team using the Scrum methodology.

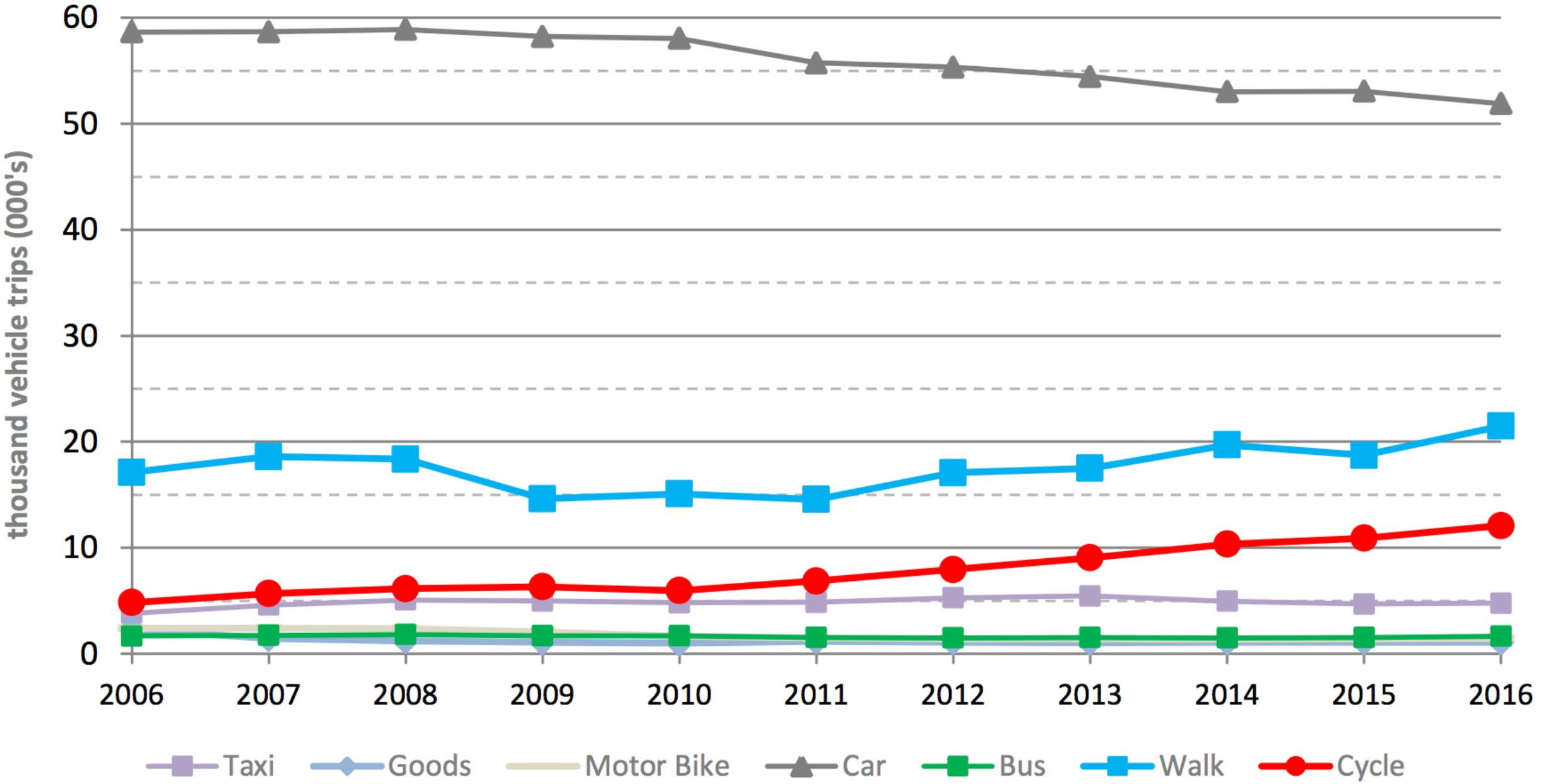
The project should be submitted before the last week of the module. This is a *hard* deadline because for the final assignment in the module, you will conduct a review of another team's project.

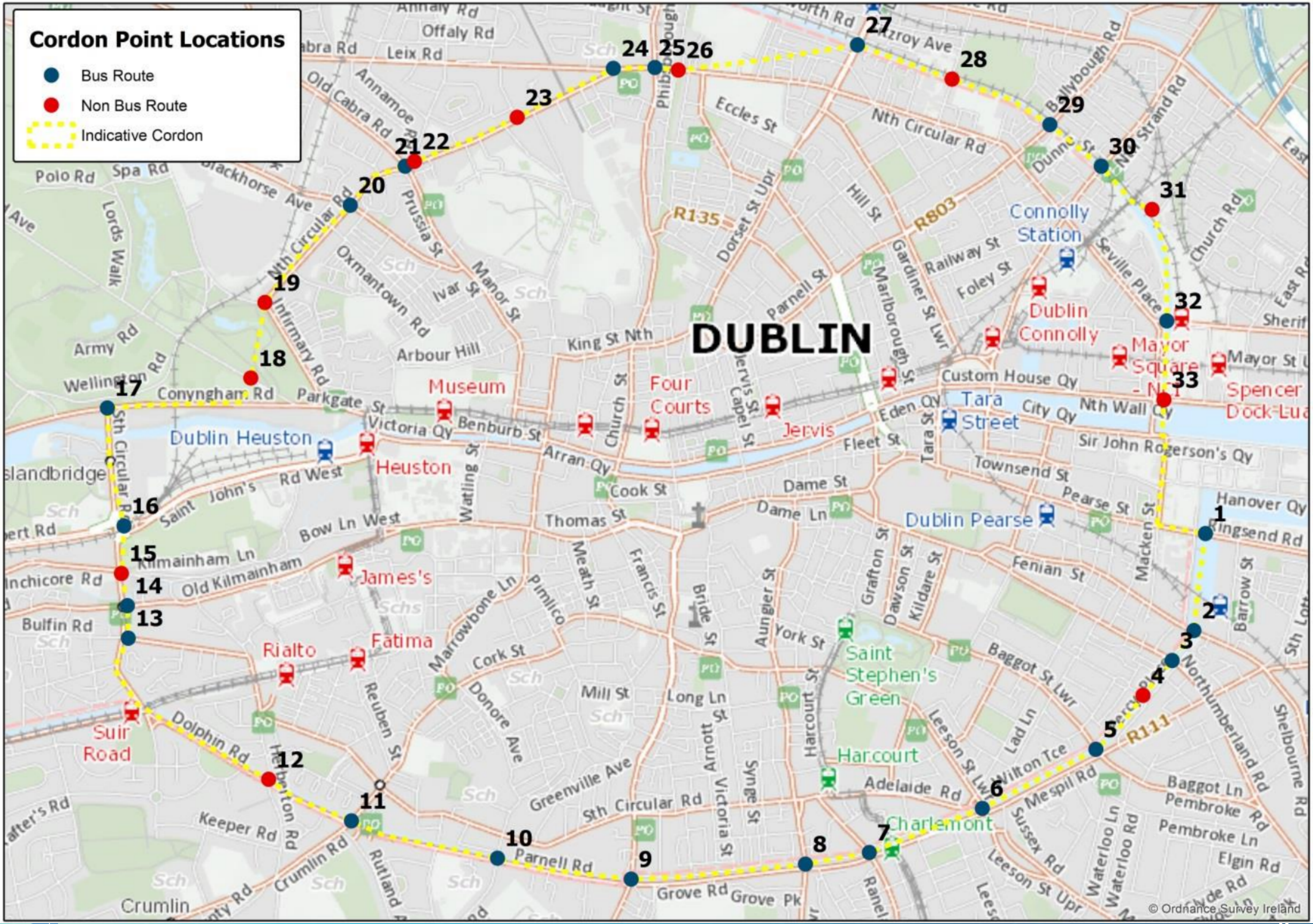


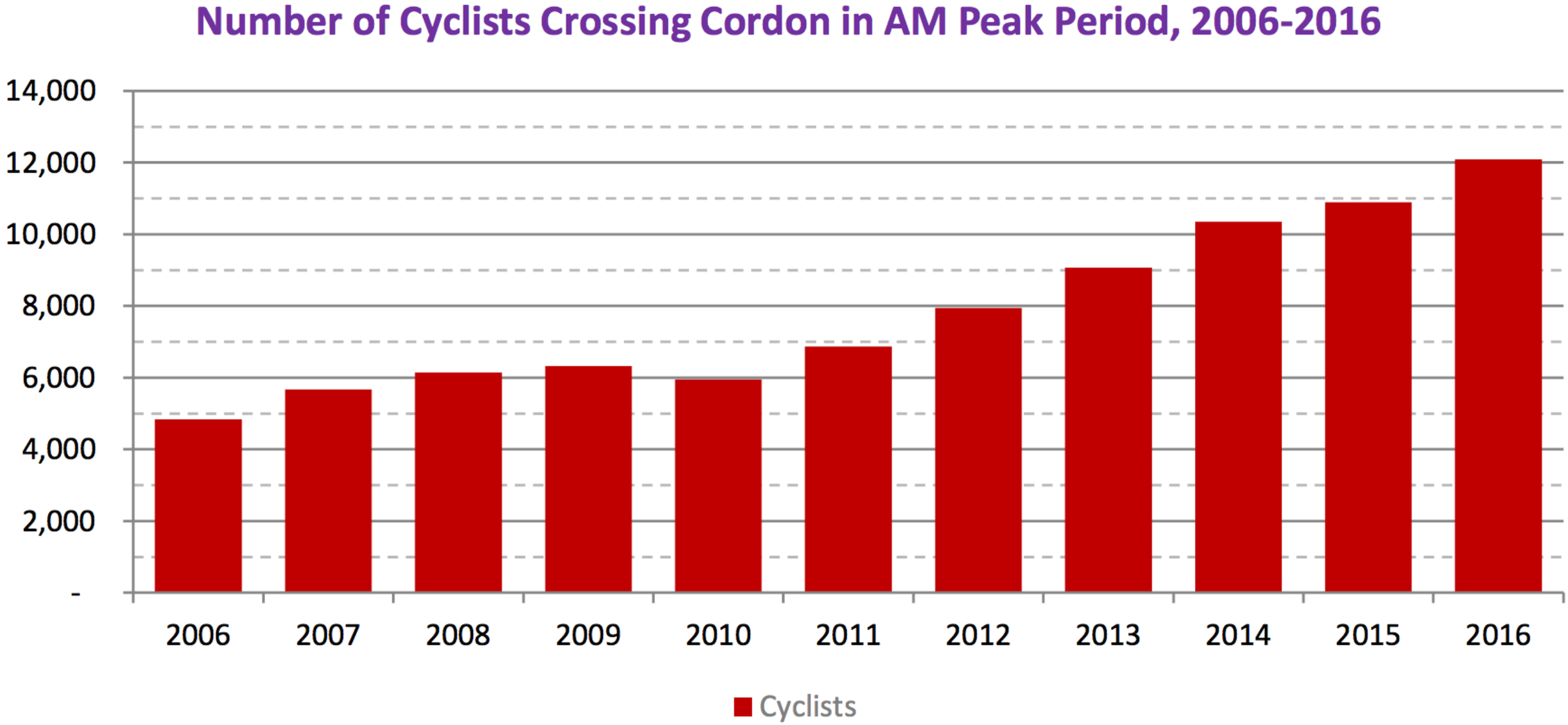
- Deliverable Project
  1. data collection through API
  2. data management/storage in DB on AWS
  3. display bike stations on map
  4. occupancy information
  5. weather information
  6. interactivity (click, API request, handle response)
  7. project served on EC2
  8. Github repo
- Project Management
  1. Scrum project management
  2. Meeting notes (log of daily standup, notes from sprint reviews, retrospectives
  3. feature selection/product backlog
  4. burndown charts, (sprint and release)
  5. managing deliverables, prototypes
- Personal Notes
  1. learning journal
  2. personal review of the project


ZE 6747

Vehicles by mode crossing the Canal Cordon 0700-1000
2006 - 2016




Cordon Point Locations
Bus Route
Non Bus Route
Indicative Cordon
DUBLIN
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
Connolly Station
Dublin Connolly
Mayor Square - NCI
Spencer Dock Lua
Tara Street
Jervis
Four Courts
Museum
Heuston
Dublin Heuston
James's
Fatima
Rialto
Suir Road
Saint Stephen's Green
Harcourt
Charlemont
Dublin Pearse
Islandbridge
Crumlin
Annaly Rd
Offaly Rd
Leix Rd
Old Cabra Rd
Annamoe Rd
Polo Rd
Spa Rd
Blackhorse Ave
Lords Walk
Army Rd
Wellington Rd
Conyngham Rd
Parkgate St
Victoria Qy
Benburb St
Arran Qy
Nth Circular Rd
Infirmary Rd
Prussia St
Oxmantown Rd
Ivar St
Manor St
Arbour Hill
King St Nth
Church St
Capel St
Jervis St
Dorset St Upr
Eccles St
Hill St
Parnell St
Marlborough St
Gardiner St Lwr
Railway St
Foley St
R135
R803
Ballybough Rd
Dunne St
Nth Strand Rd
Seville Place
Church Rd
East Rd
Sheriff
Mayor St
Custom House Qy
City Qy
Nth Wall Qy
Eden Qy
Tara St
Sir John Rogerson's Qy
Fleet St
Townsend St
Pearse St
Hanover Qy
Ringsend Rd
Macken St
Fenian St
Barrow St
Sth Circular Rd
Saint John's Rd West
Bow Ln West
Watling St
Cook St
Thomas St
Dame St
Dame Ln
Kilmainham Ln
Old Kilmainham
Inchicore Rd
Bulfin Rd
Meath St
Francis St
Marrowbone Ln
Pimlico
Cork St
Bride St
Aungier St
York St
Grafton St
Dawson St
Kildare St
Baggot St Lwr
Northumberland Rd
Shelbourne Rd
Reuben St
Donore Ave
Mill St
Long Ln
Harcourt St
Leeson St Lwr
Lad Ln
Wilton Tce
Mespil Rd
R111
Dolphin Rd
Herberton Rd
Greenville Ave
Sth Circular Rd
Arnott St
Victoria St
Synge St
Adelaide Rd
Sussex Rd
Leeson St Upr
Waterloo Ln
Waterloo Rd
Baggot Ln
Pembroke Rd
Pembroke Ln
Elgin Rd
Clyde Rd
Keeper Rd
Crumlin Rd
Rutland
Parnell Rd
Grove Rd
Grove Pk
Ranelagh
© Ordnance Survey Ireland

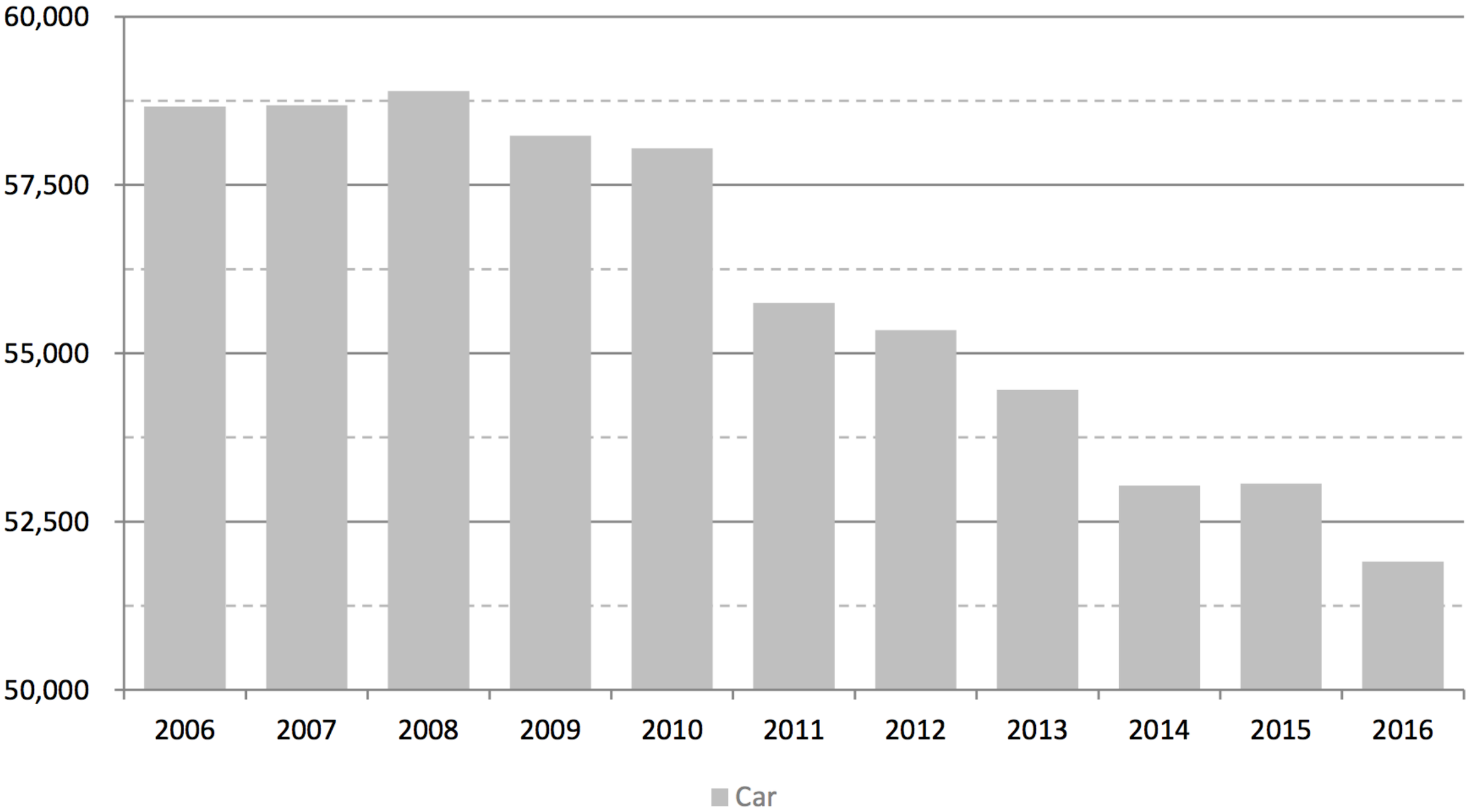


2.2.5 Cyclists

There was an increase of 11% in the number of cyclists crossing the canal cordon from 2015 to 2016. There has been a steady year on year growth in the number of cyclists crossing the cordon since 2010. In 2016 over 12,000 cyclists crossed the cordon in the AM peak period. This represents an increase of 150% when compared with 2006, and represents an increase of over 50% in the last four years.

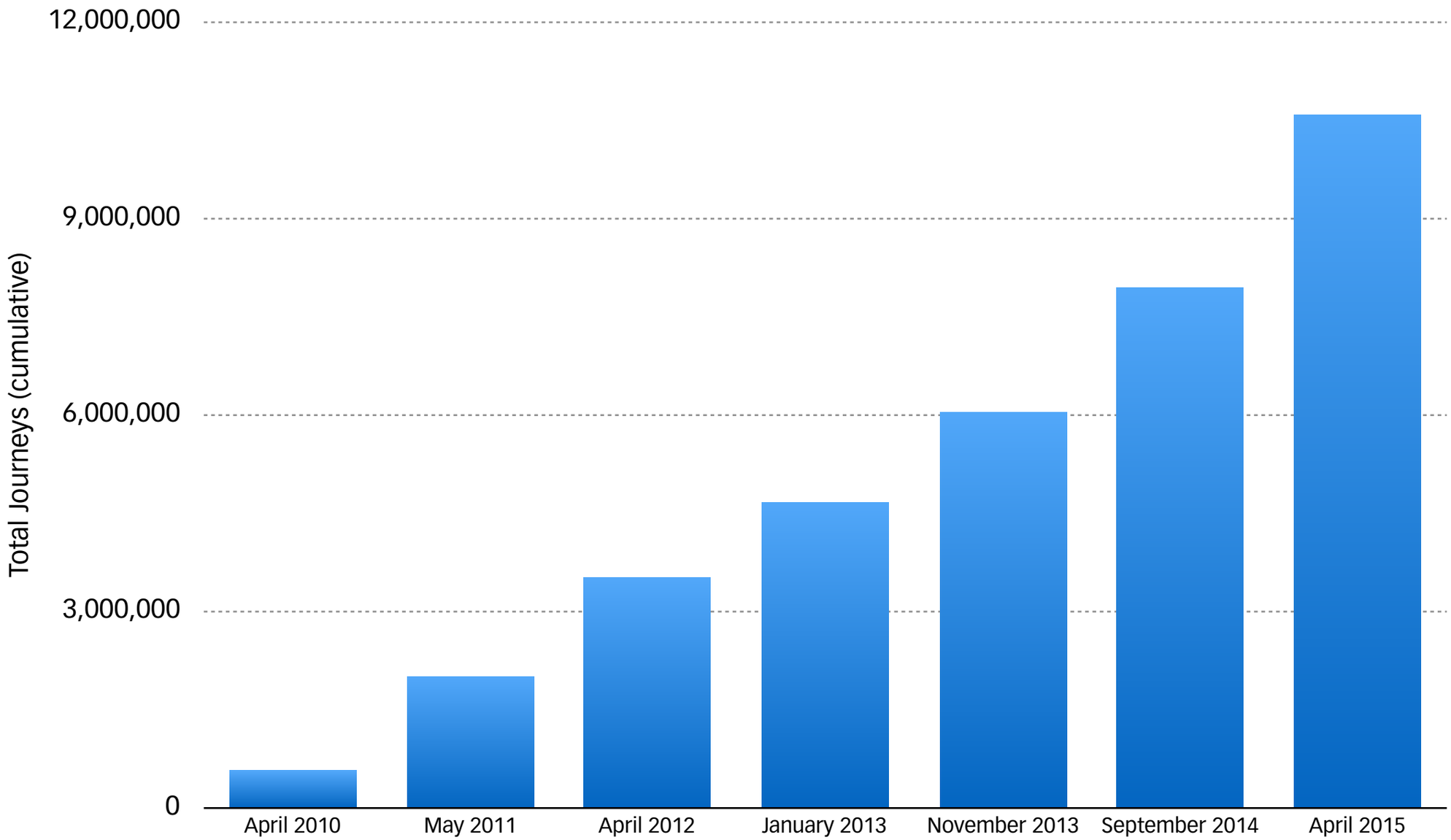
In the period 2006-2016 the peak year for cars crossing the canal cordon was in 2008 with almost 59,000 vehicles. The 2016 Figurerepresents a decrease of 12% or 6,989 cars since this peak.

Number of Cars Crossing Cordon in AM Peak Period, 2006-2016




dublinbikes
00324

- 1500 bikes
- 102 stations
- 55k subscribers
- launched in Sept. 2009



Current Valid Long Term Subscribers: 68,805

Short Term Subscribers (YTD): 646

Journeys (YTD): 325, 331

Journeys (since launch): 18,095,097

Average Duration of Journey (YTD): 14 minutes

Percentage of Journeys Free i.e. under 30 minutes (YTD): 98%

Busiest Usage Day Ever: 16th September 2016

Number of Journeys on Busiest Day: 18,041

dublinbikes.ie

Station map

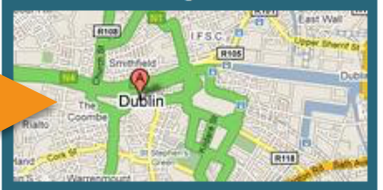
Cookie Disclosure
This website uses cookies. By continuing to use the website, you consent to the use of cookies. Read more .
OK

Coca-Cola zero dublinbikes

Comhairle Cathrach Bhaile Átha Cliath
Dublin City Council

HOME | STATIONS | HOW DOES IT WORK? | SUBSCRIPTION | MAGAZINE | SAFETY | CONTACT | EXPANSION NEWS

JOIN THE SCHEME
Click here to subscribe to Coca-Cola Zero dublinbikes
ANNUAL CARD

Station Map


Address, key word

List of stations
Click here for the full list of Coca-Cola Zero dublinbikes stations and to download the map.
VIEW STATIONS

Consult my account
My account

View the Terms & Conditions

Coca-Cola Zero dublinbikes – Update 29th Jan 2016
All Coca-Cola Zero dublinbikes stations are now fully reconnected following a technical issue earlier today.
Read more
NEWS

How does it work?
All you need to know about Coca-Cola Zero dublinbikes, the bike sharing scheme for Dublin city.
HOW DOES IT WORK?

Coca-Cola Zero dublinbikes FAQ's
What happens if a station is full and I can't return my bike?
How do I lock the bike?
How do I renew or cancel my Coca-Cola Zero dublinbikes subscription?
How do I contact the call centre from outside of Ireland?
MORE FAQ'S

Contact Us
For telephone queries please contact 1850 777 070 or 01 424 7605, for Email queries please click below
CONTACT

Road Safety Guidelines
Some safety recommendations and helpful tips to travel with complete peace of mind around the city on Coca-Cola Zero dublinbikes.
FIND OUT MORE

Coca-Cola Zero dublinbikes is expanding!

→ new stations

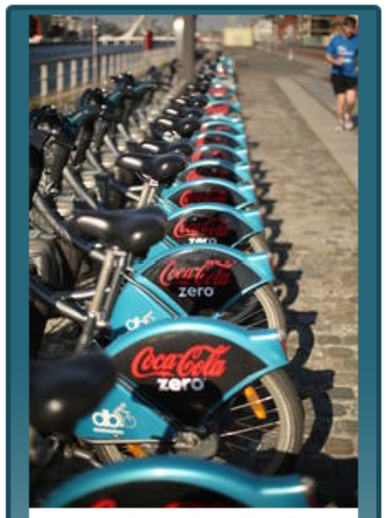


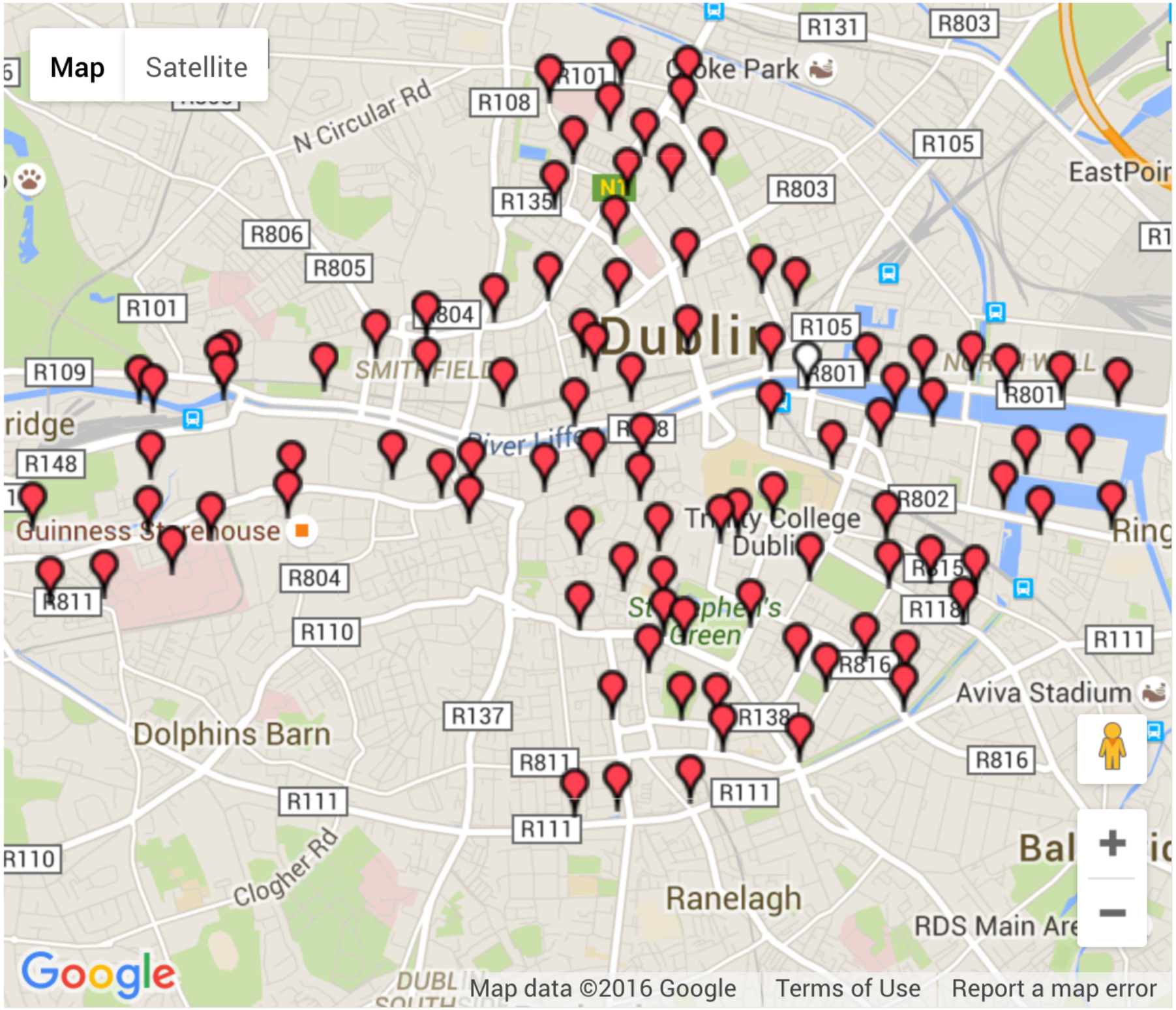
Image Credit: Coca-Cola Zero dublinbikes user Bazphoon.

Have you got a great photo of Coca-Cola Zero dublinbikes that you would like to share with us?

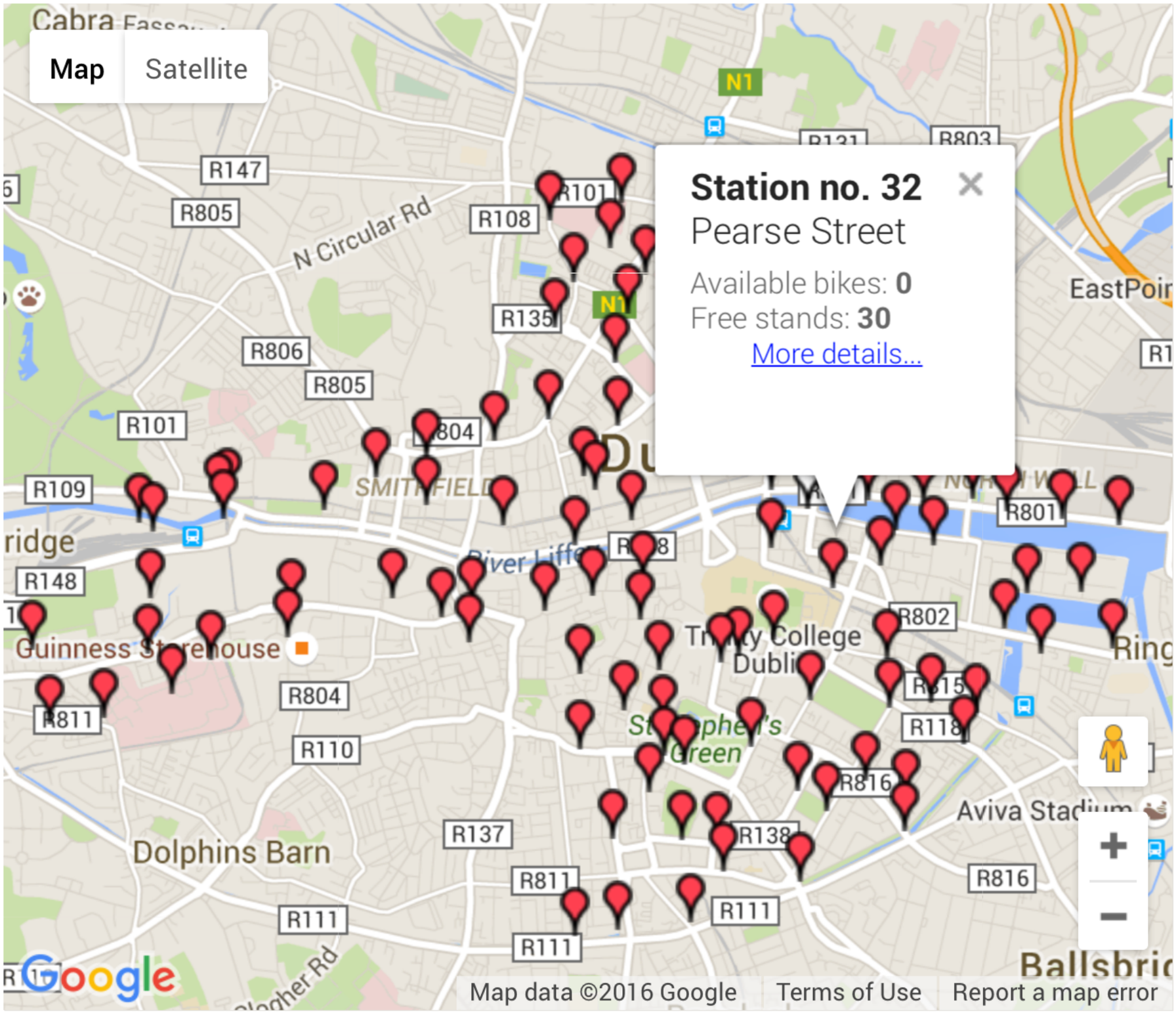
Email it to photos@dublinbikes.ie

Stations Open: 05.00am - 00.30am
Return a bike: 24 hours

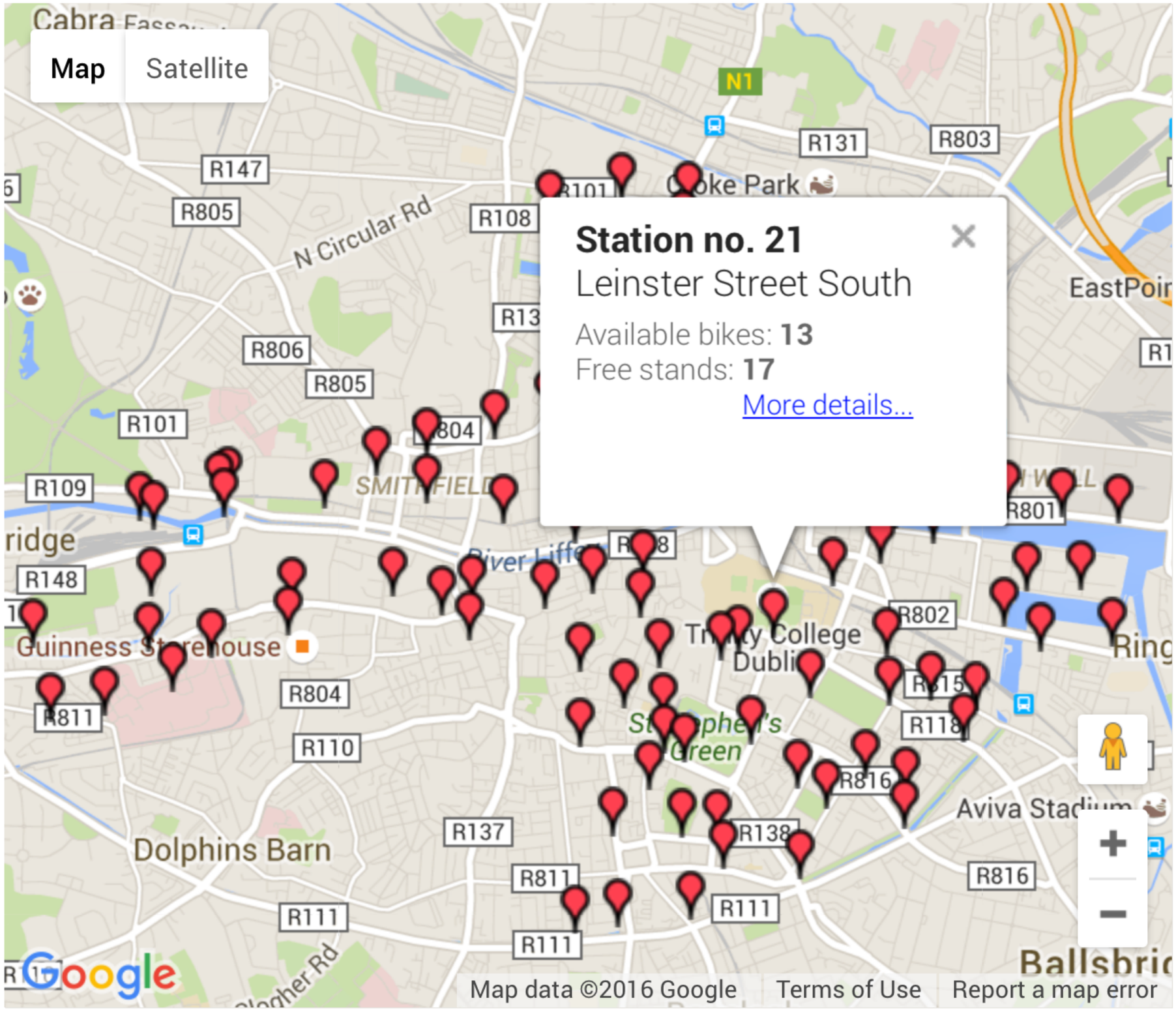
Station map


Map
Satellite
N Circular Rd
R108
R101
R131
R803
R105
R135
N1
R806
R805
R804
R109
R148
R811
R110
R137
R138
R816
R802
R801
R111
Dublin
SMITHFIELD
Guinness Storehouse
Trinity College
Dolphins Barn
Ranelagh
Clogher Rd
Aviva Stadium
Google
Map data ©2016 Google
Terms of Use
Report a map error

Station map


Map
Satellite
Station no. 32
Pearse Street
Available bikes: 0
Free stands: 30
More details...
N1
R147
R805
N Circular Rd
R108
R101
R135
R806
R804
R109
R148
R811
R110
R137
R138
R111
R816
R801
R802
R803
R815
R118
Guinness Storehouse
Trinity College
Dolphins Barn
Aviva Stadium
Google
Map data ©2016 Google
Terms of Use
Report a map error

Station map


Map
Satellite
Station no. 21
Leinster Street South
Available bikes: 13
Free stands: 17
More details...
N1
R131
R803
R147
R805
N Circular Rd
R108
R806
R101
R109
R148
R804
R811
R110
R137
R111
R802
R815
R118
R816
R138
Guinness Storehouse
Dolphins Barn
SMITHFIELD
Trinity College
Aviva Stadium
Google
Map data ©2016 Google
Terms of Use
Report a map error

- basic station information

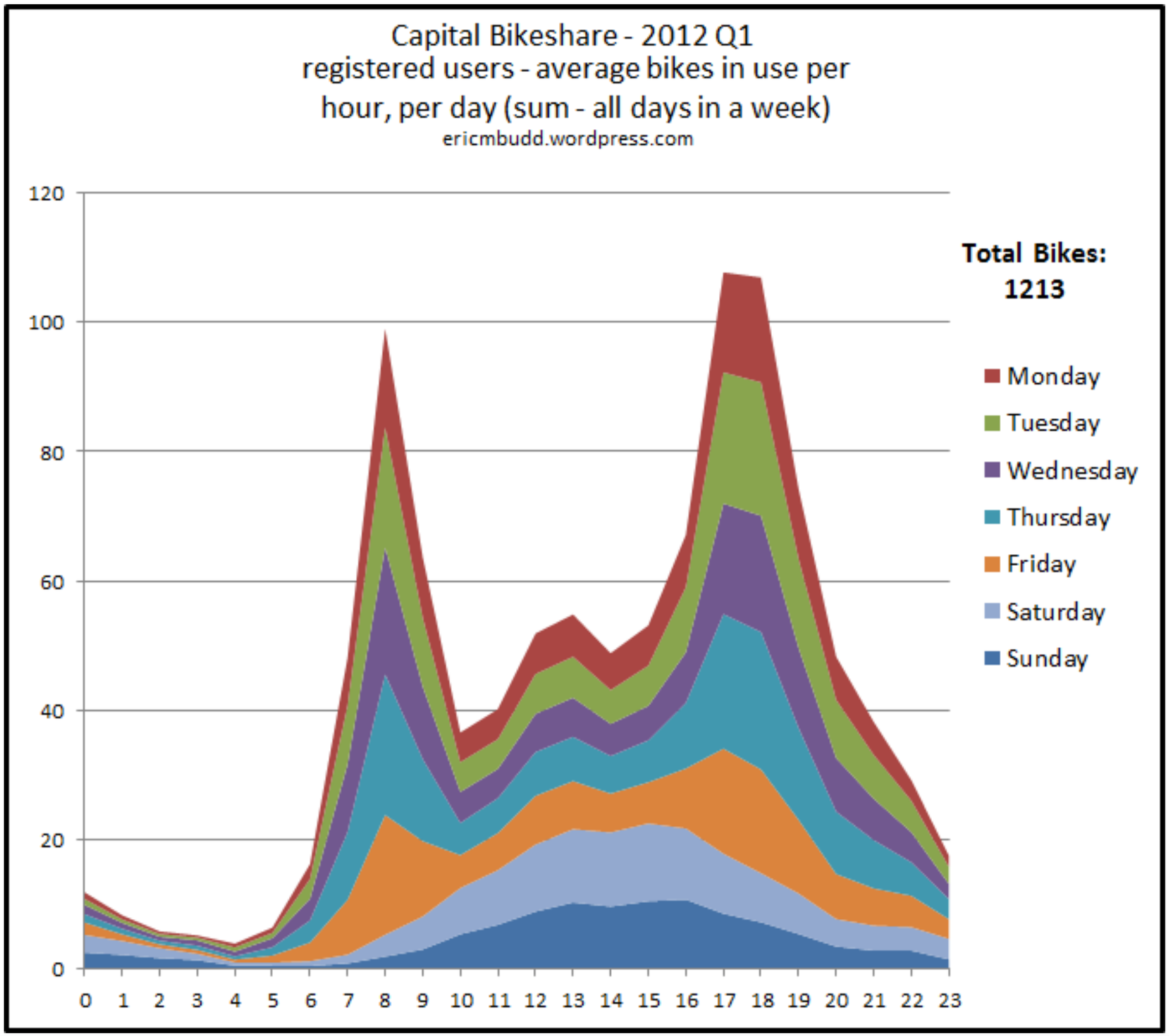
Your search

Station no. 21

Full address:
Leinster Street South

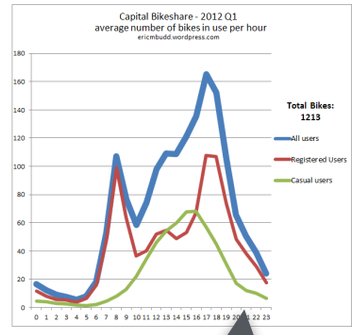
Available bikes: 13
Free stands: 17
Total capacity: 30
Credit cards accepted: NO

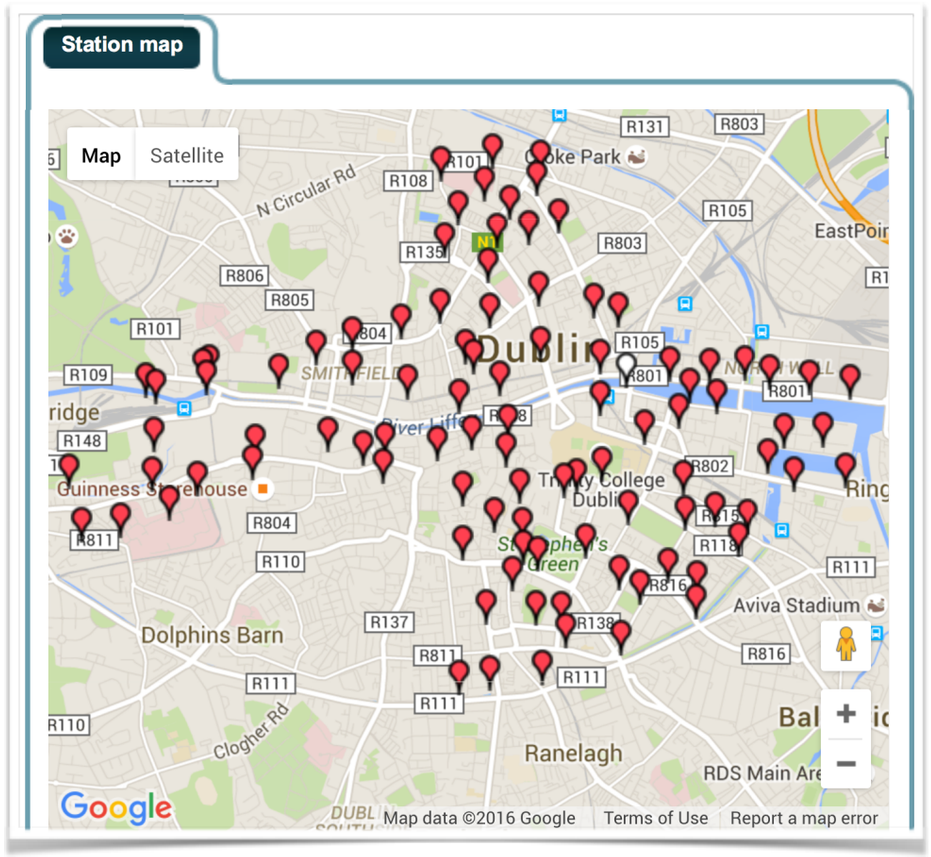
- demand various by hour of the day
- and by day of the week

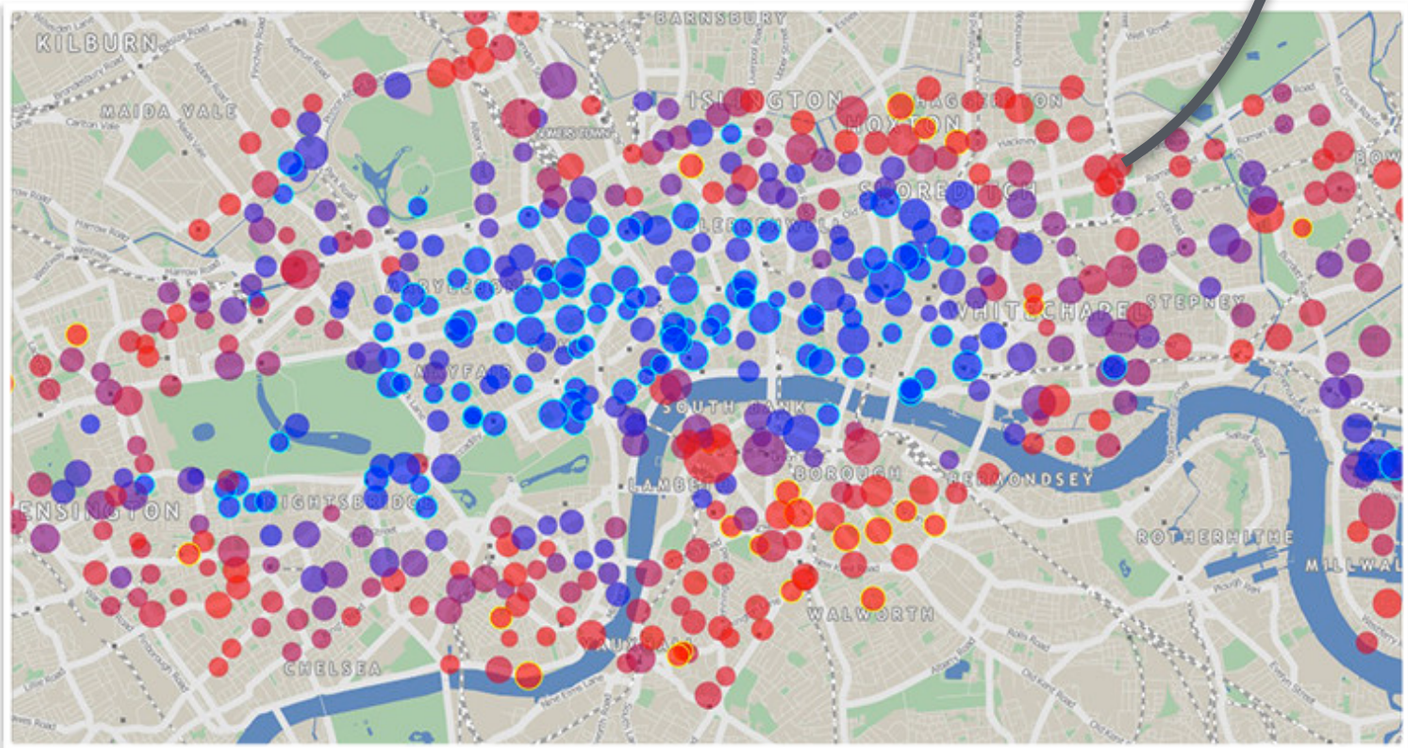


Project Task:

Add occupancy information to Dublin Bikes station map

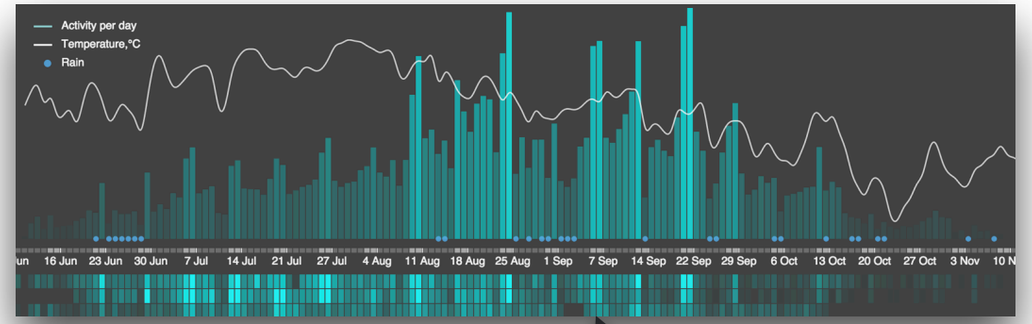


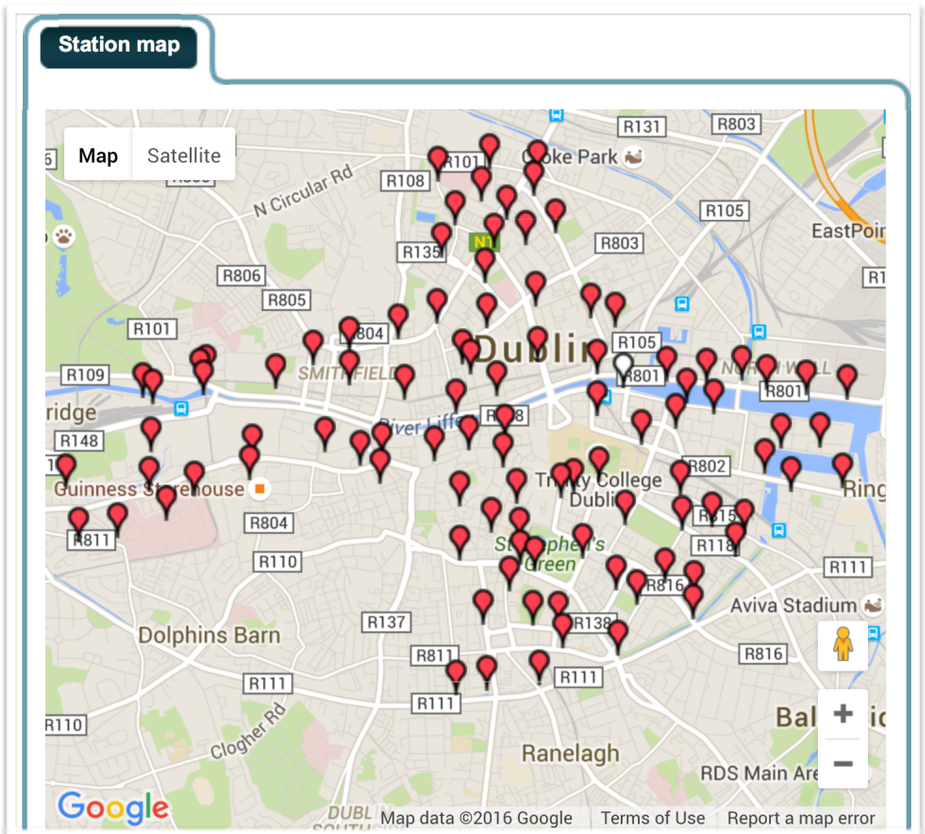


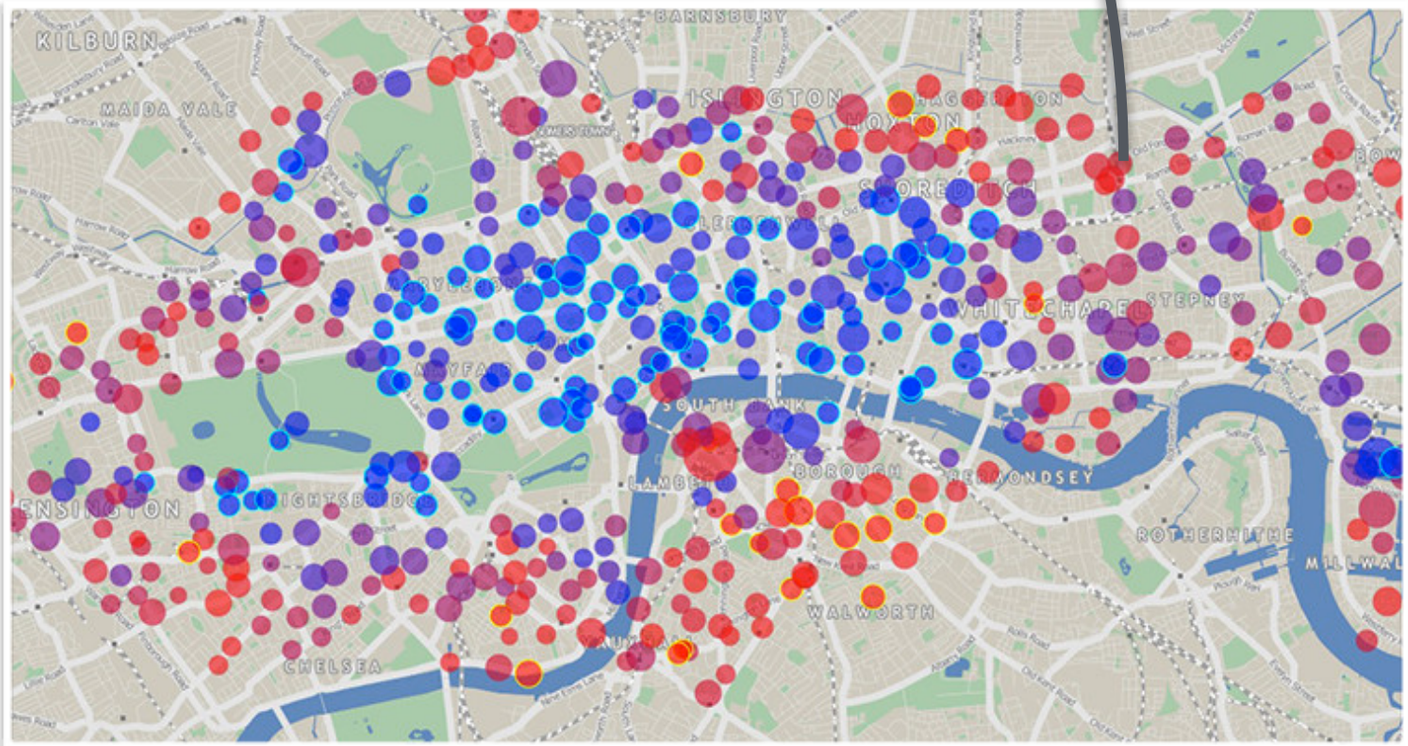


Project Task:

Add weather information

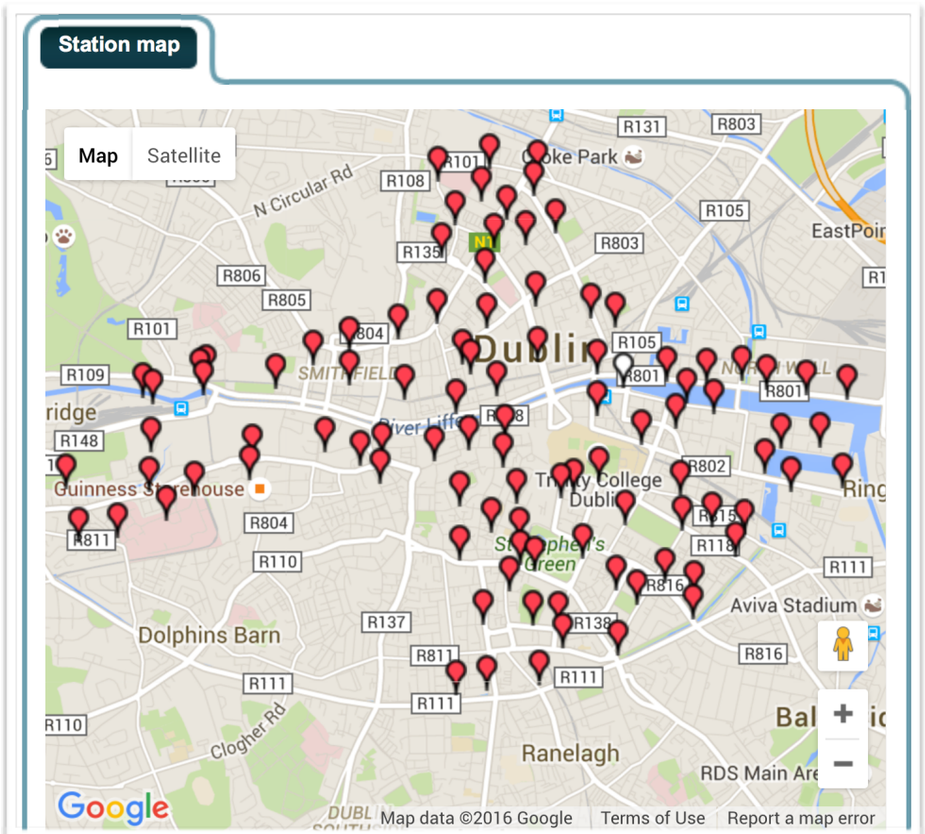


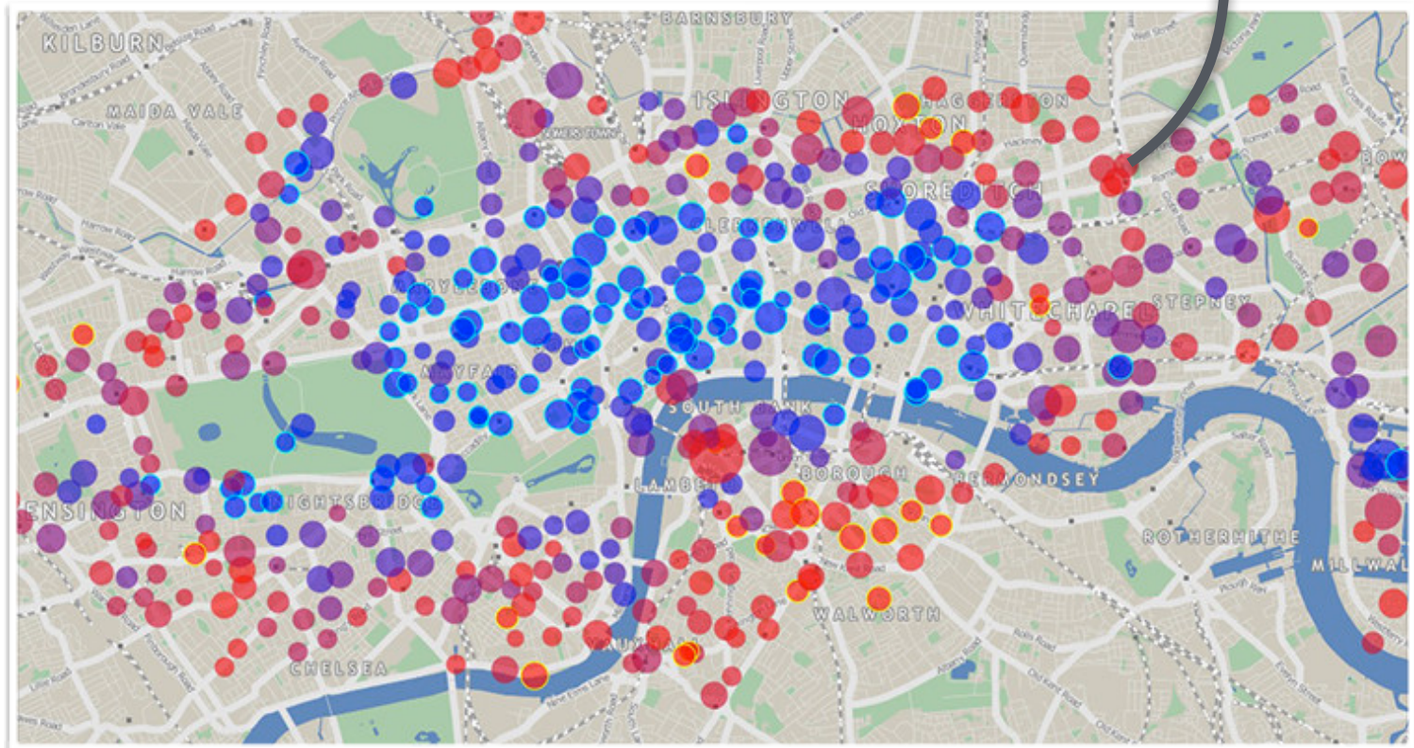




Project Task:

Add weather information





Project Task

- **mine "DublinBikes" dynamic station occupancy data from JCDecaux**
  - every 5 mins, aim to have at least 1 week of continuous data
- **store the data in a data base**
  - SQLite is fine, AWS S3, other dbs...
- **display the station data on a google map**
  - encode the occupancy and availability in the colour/size of the markers
- **simple map interactivity**
  - display more detailed occupancy (hourly, daily) bar chart when a station is clicked
  - click on station should display the weather forecast (openweather

Project Task

- **At your first meeting you will need to decide on a project plan and figure out the features you will implement.**

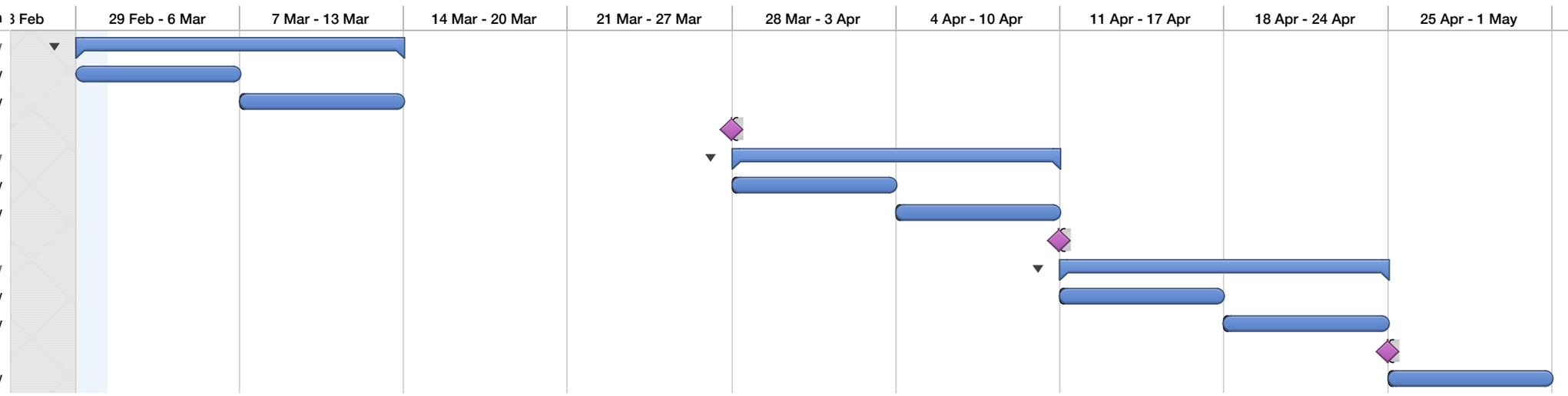
- **Specification, use cases**
- **discuss what data you need to collect**
- **plan how to collect the data**
- **set up the collection and storage**
- **process the data and make it ready for display**
- **interpret, discuss and refine**

- **sample project plan:**
- **you should invite a demonstrator to your sprint review meetings**

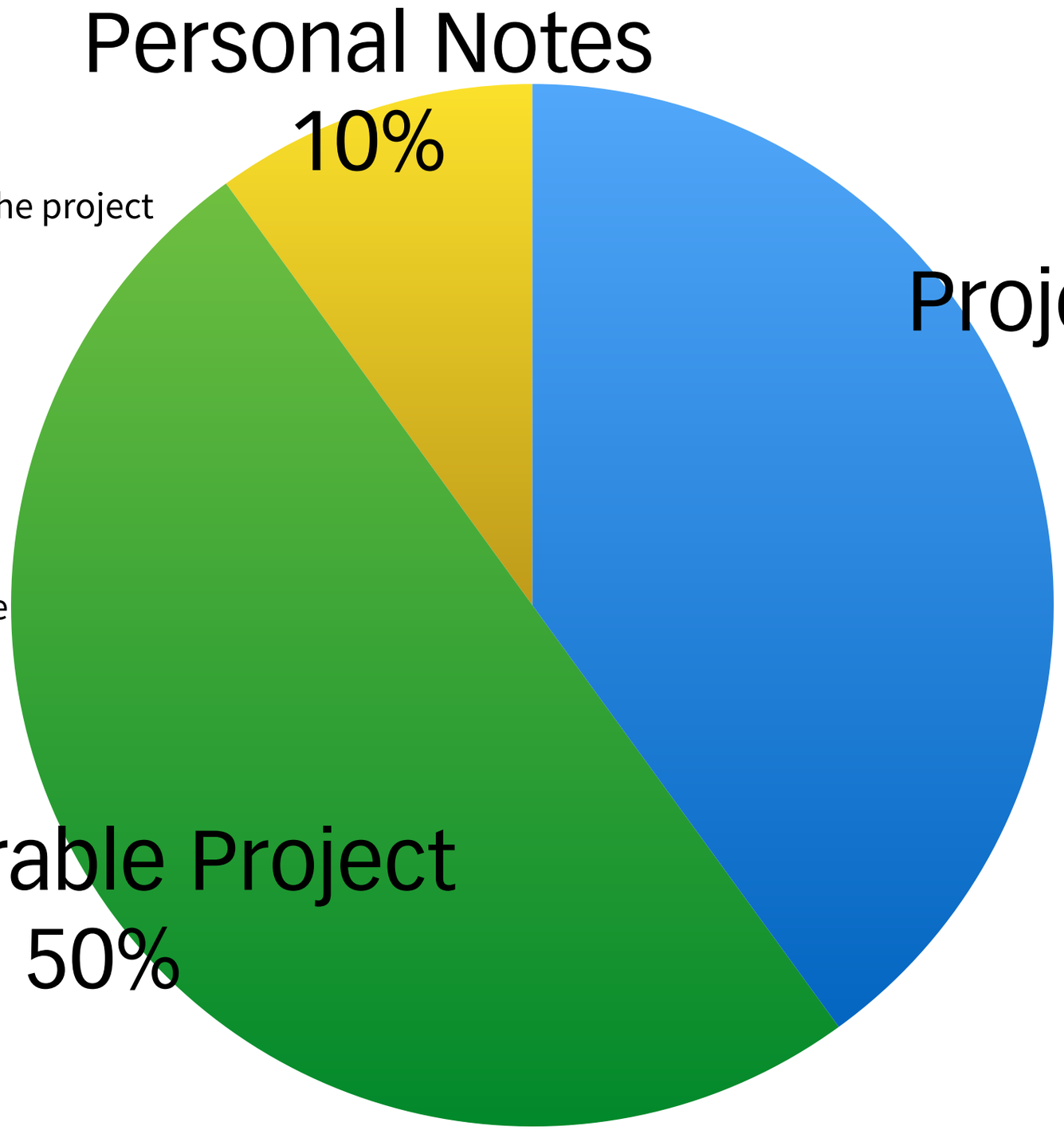
Final Deadline:
next-to-last week
of module
Friday 22nd April

sprint 1 sprint 2 sprint 3 reviews

Title	Effort	Duration
1) Sprint 1	2w	2w
• 1.1) Week 1	1w	1w
• 1.2) Week 2	1w	1w
◆ 2) Sprint Review 1		
3) Sprint 2	2w	2w
• 3.1) Week 3	1w	1w
• 3.2) Week 4	1w	1w
◆ 4) Sprint Review 2		
5) Sprint 3	2w	2w
• 5.1) Week 5	1w	1w
• 5.2) Week 6	1w	1w
◆ 6) Sprint Review 3		
• 7) Code Review	1w	1w



Marking



- learning journal
- your own review of the project

- data collection
- data management/storage
- display on map
- occupancy calculation
- weather
- interactivity

- project management
- meetings
- feature selection/ product backlog
- burndown charts
- managing deliverables

Data

- where does the data come from?

- the Dublin Bike scheme is sponsored by JCDecaux, a multinational advertising company which runs bike sharing schemes in many cities
- 102 stations and 1500 bikes
- open API to access station and availability data

Data

- **where does the data come from?**

Two kinds of data are delivered by the platform:

Static data provides stable information like station position, number of bike stands, payment terminal availability, etc.

can be downloaded manually in file format or accessed through the API

Dynamic data provides station state, number of available bikes, number of free bike stands, etc.. Refreshed every minute and can be accessed only through the API

JCDecaux developer

EN | FR

home open data contact sign in

Open Data

Enjoy our open data to create innovative services.

We believe in shared innovation and the creative potential of communities to make cities ever more inventive and accessible. With JCDecaux developer, create new applications and services through an easy-to-access distibution of data under « Open License ».

JCDecaux's self-service bicycles

Access our bike data now!

From the location of the bike stations to the availability of bikes and parking spaces in real time, use our data to experiment new representations or to provide innovative and useful services to users. You can access this data through a simple download or an advanced web API.

jcdecaux.com

cyclocity.com

@jcdecauxdev

developer@jcdecaux.com

- **Static data doesn't require an API key.**
- **For access to dynamic data, you must use a personal API key.**

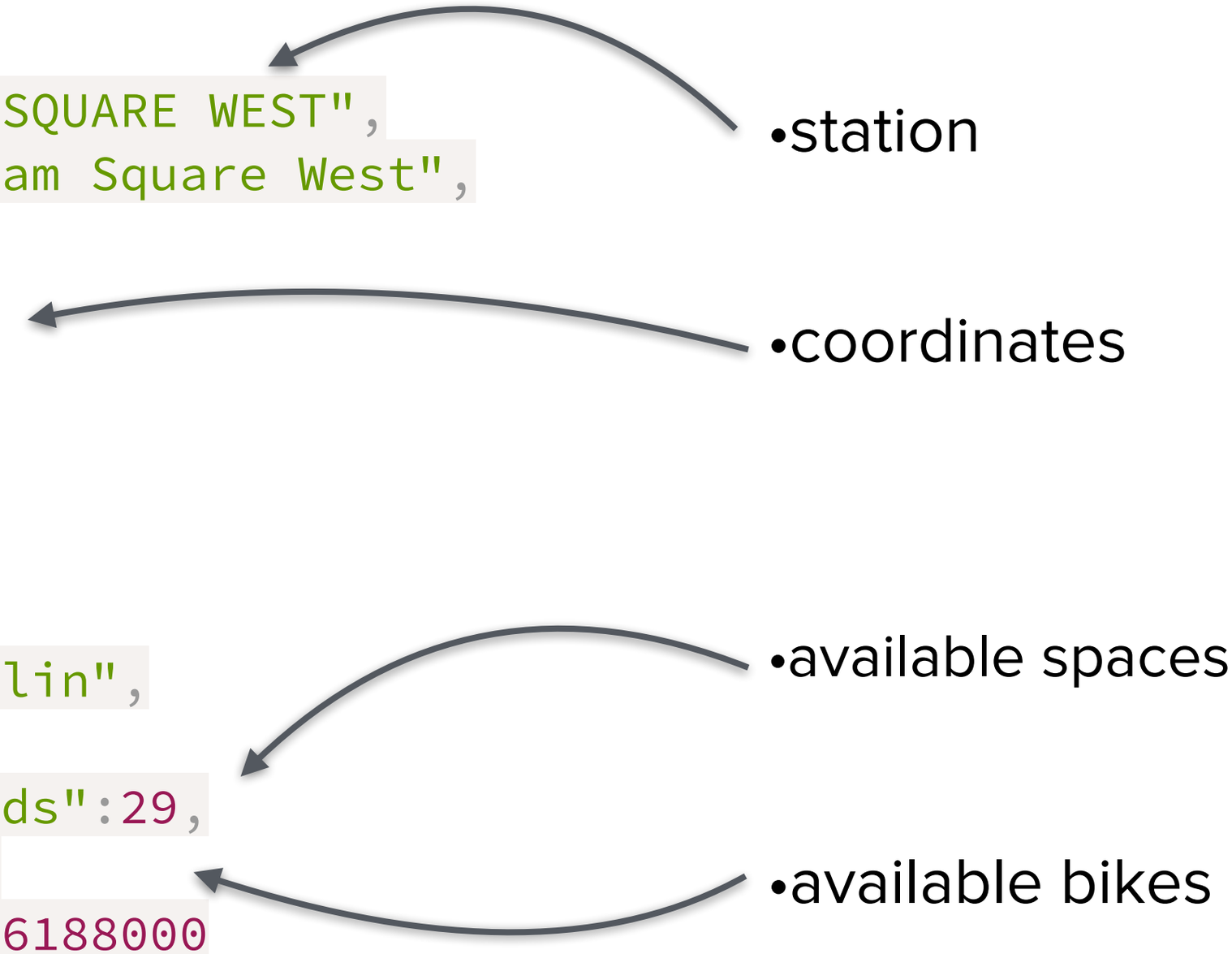
Once you have a key, you just have to add it as a parameter named « apiKey » to all your requests to the dynamic API.

GET https://api.jcdecaux.com/vls/v1/stations?contract={contract_name}&apiKey={api_key}

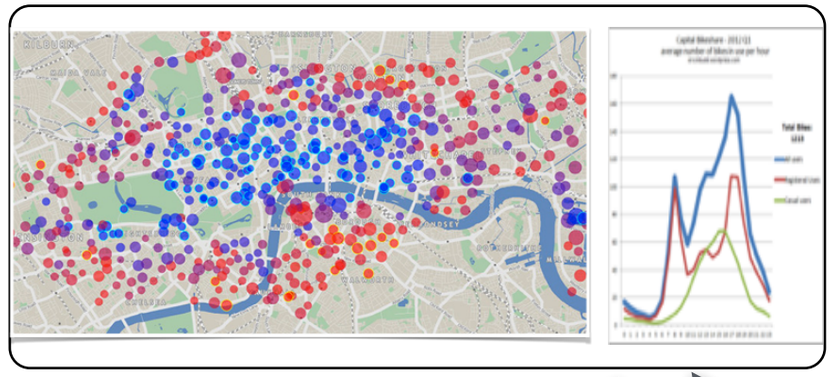
If no key is specified, you'll get a `403 Forbidden` error.

- **sample JSON response:**


```
{
      "number":13,
      "name":"FITZWILLIAM SQUARE WEST",
      "address":"Fitzwilliam Square West",
      "position":{
         "lat":53.336074,
         "lng":-6.252825
      },
      "banking":false,
      "bonus":false,
      "status":"OPEN",
      "contract_name":"Dublin",
      "bike_stands":30,
      "available_bike_stands":29,
      "available_bikes":1,
      "last_update":1456696188000
   }
```

client



requests

JCDecaux developer

Python code to scrape station and occupancy data from online source

Data Store

store data in a database (SQLite or MongoDB)

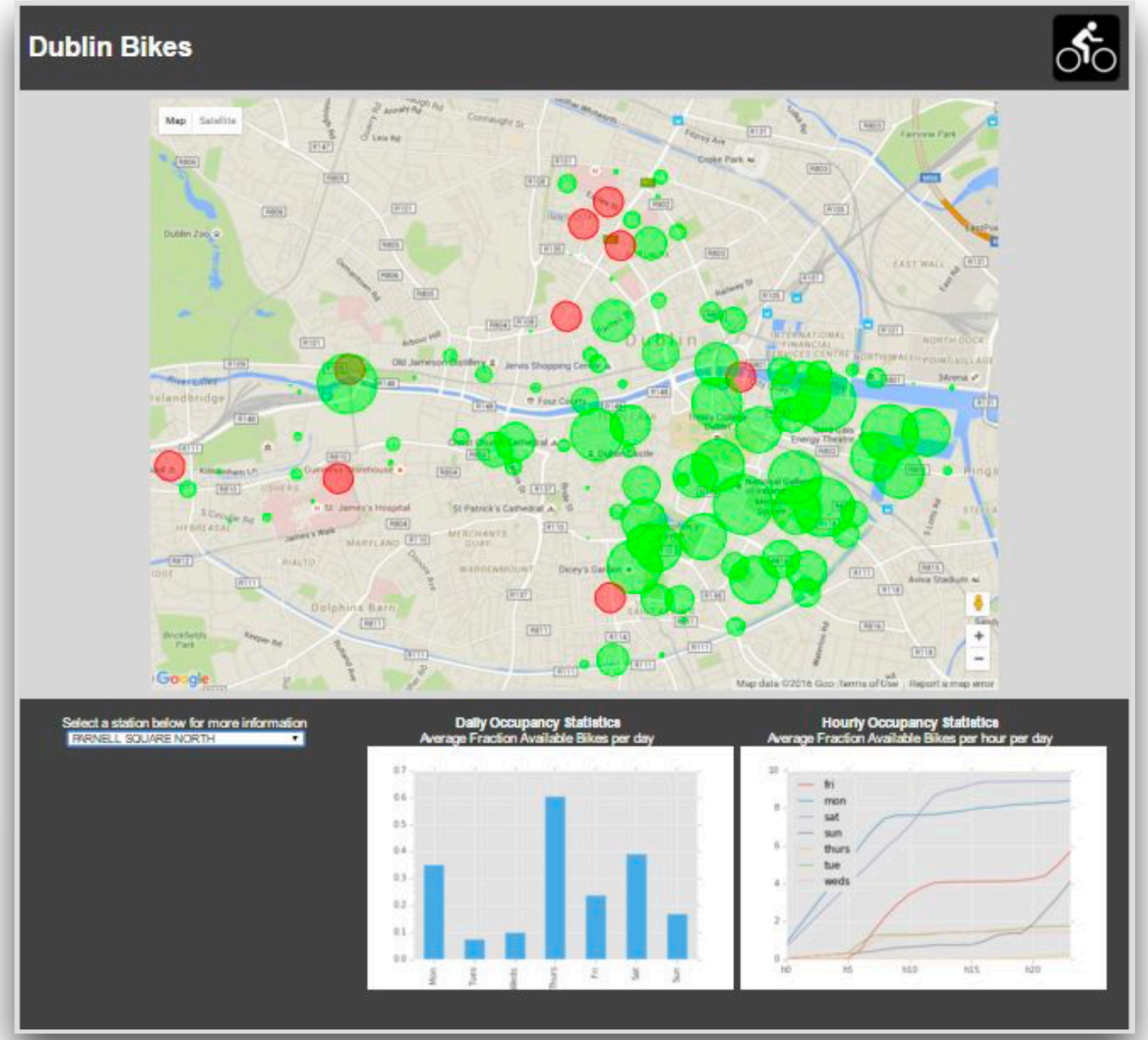
Python code to analyse data and provide occupancy information

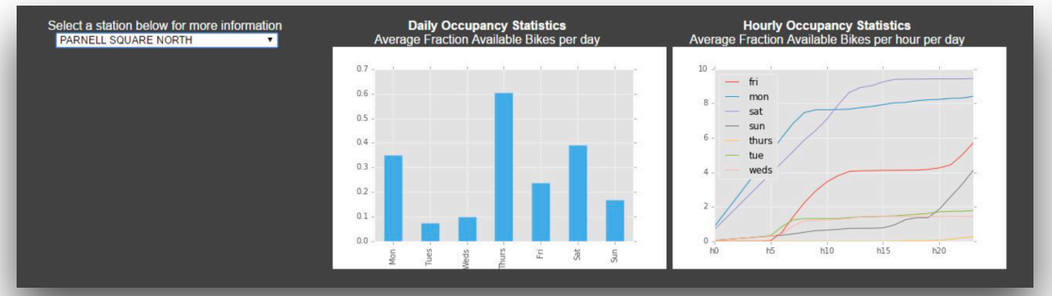
server

AWS

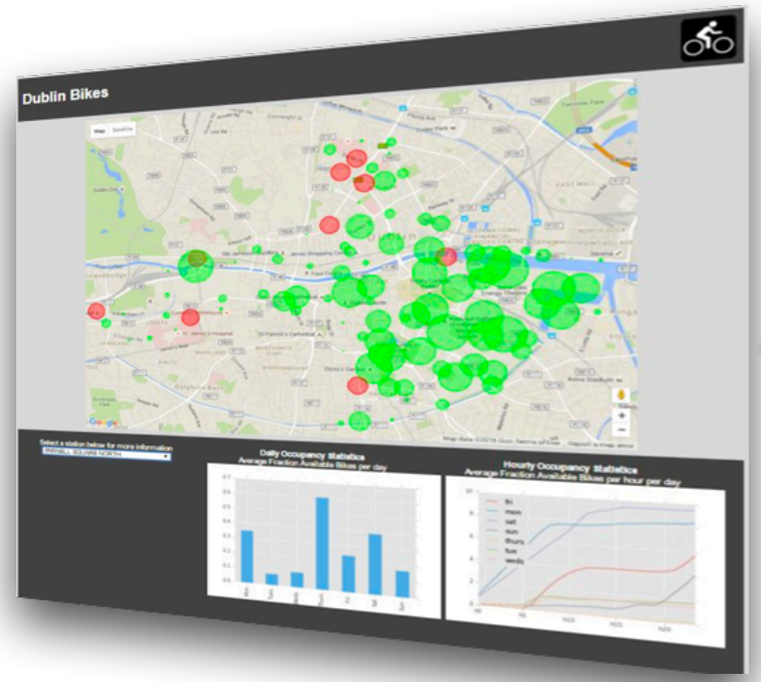
COMP30670 Project

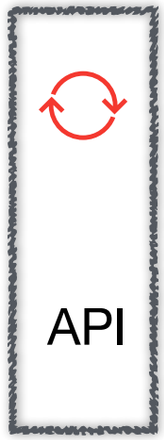
work samples





COMP30670 Project

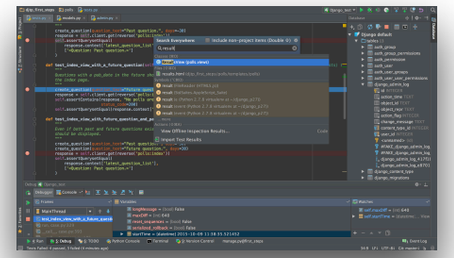




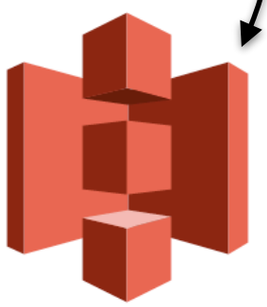
API's are implemented with python *flask* app

push to github

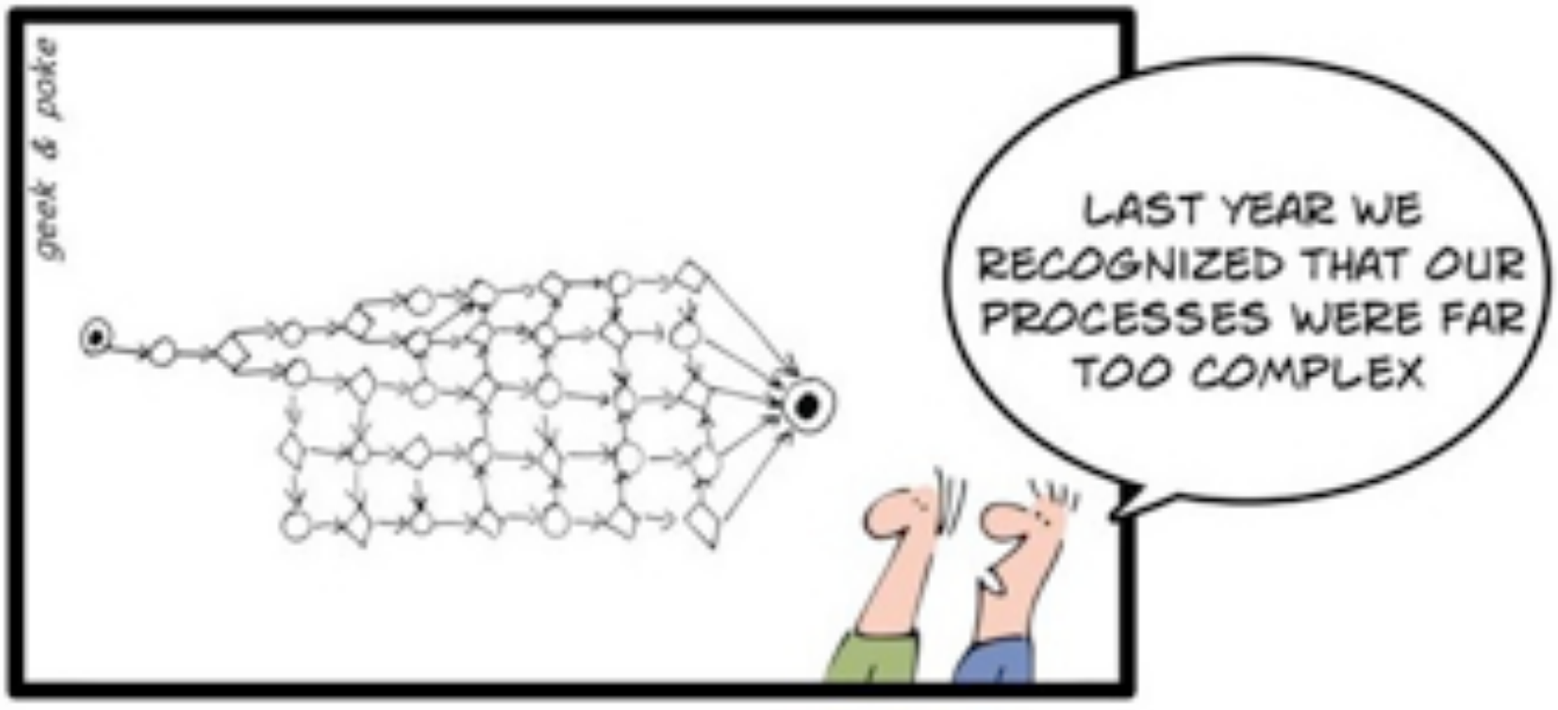
pull from github

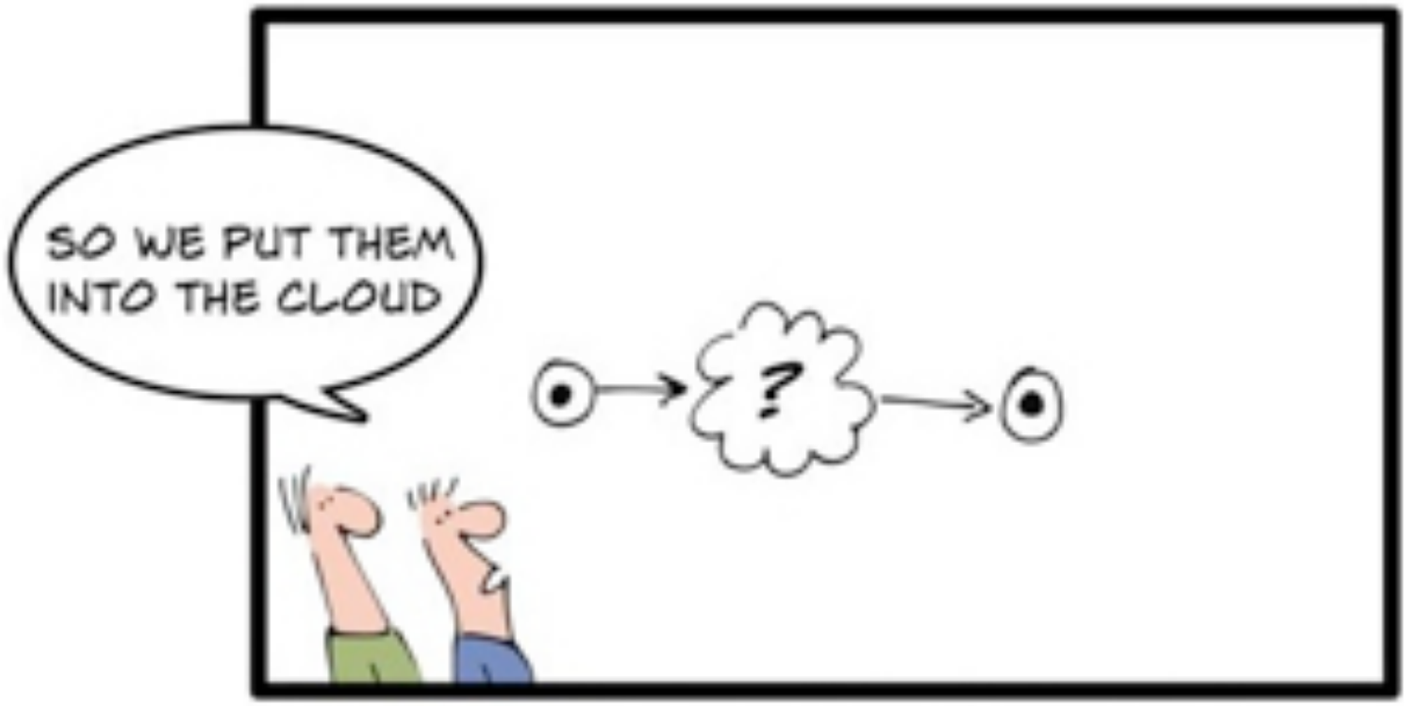


develop on your local machine



store in S3





LET THE CLOUDS MAKE YOUR LIFE EASIER

{JSON}

amazon web services | EC2

Requests

Requests is an elegant and simple HTTP library for Python, built for human beings.

Donate

If you love Requests, consider supporting the author on Gittip:

Get Updates

Receive updates on new releases and upcoming projects.

Subscribe to Newsletter

Translations

Requests: HTTP for Humans

Release v2.3.0. (Installation)

Requests is an Apache2 Licensed HTTP library, written in Python, for human beings.

Python's standard **urllib2** module provides most of the HTTP capabilities you need, but the API is thoroughly **broken**. It was built for a different time — and a different web. It requires an *enormous* amount of work (even method overrides) to perform the simplest of tasks.

Things shouldn't be this way. Not in Python.

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type":"User"...'
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

See similar code, without Requests.

python™

SQLite

Flask

web development, one drop at a time

Google Maps

Google Developers CHARTS

Advice

My role:

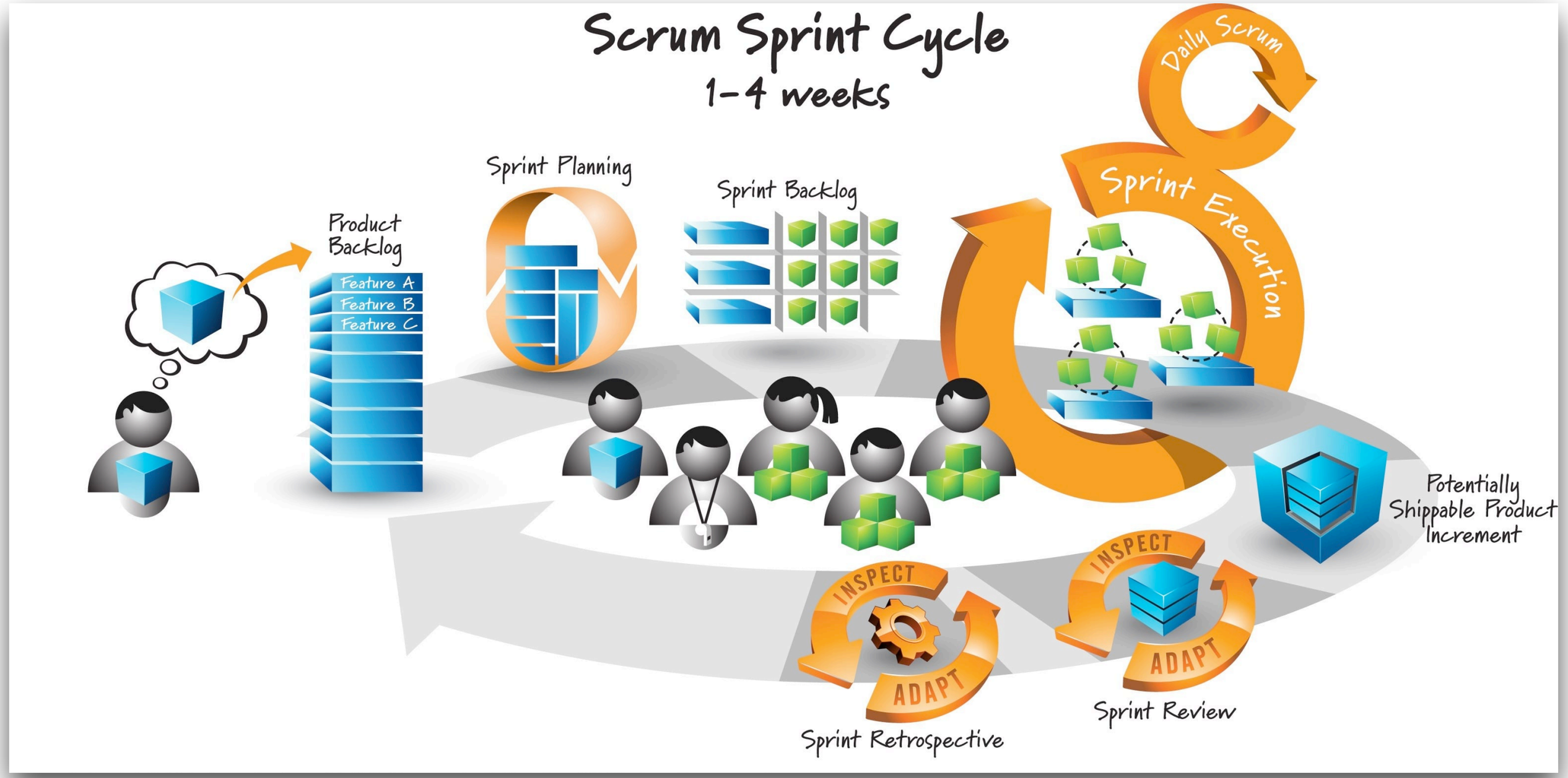
Me (or demonstrator) = Product Owner

- Your group is the Development Team
- Each team will be assigned to a demonstrator (or myself) who will act as product owner
- The product owner will advise you as you develop the product backlog
- You should schedule meetings with the product owner to discuss and refine the backlog.

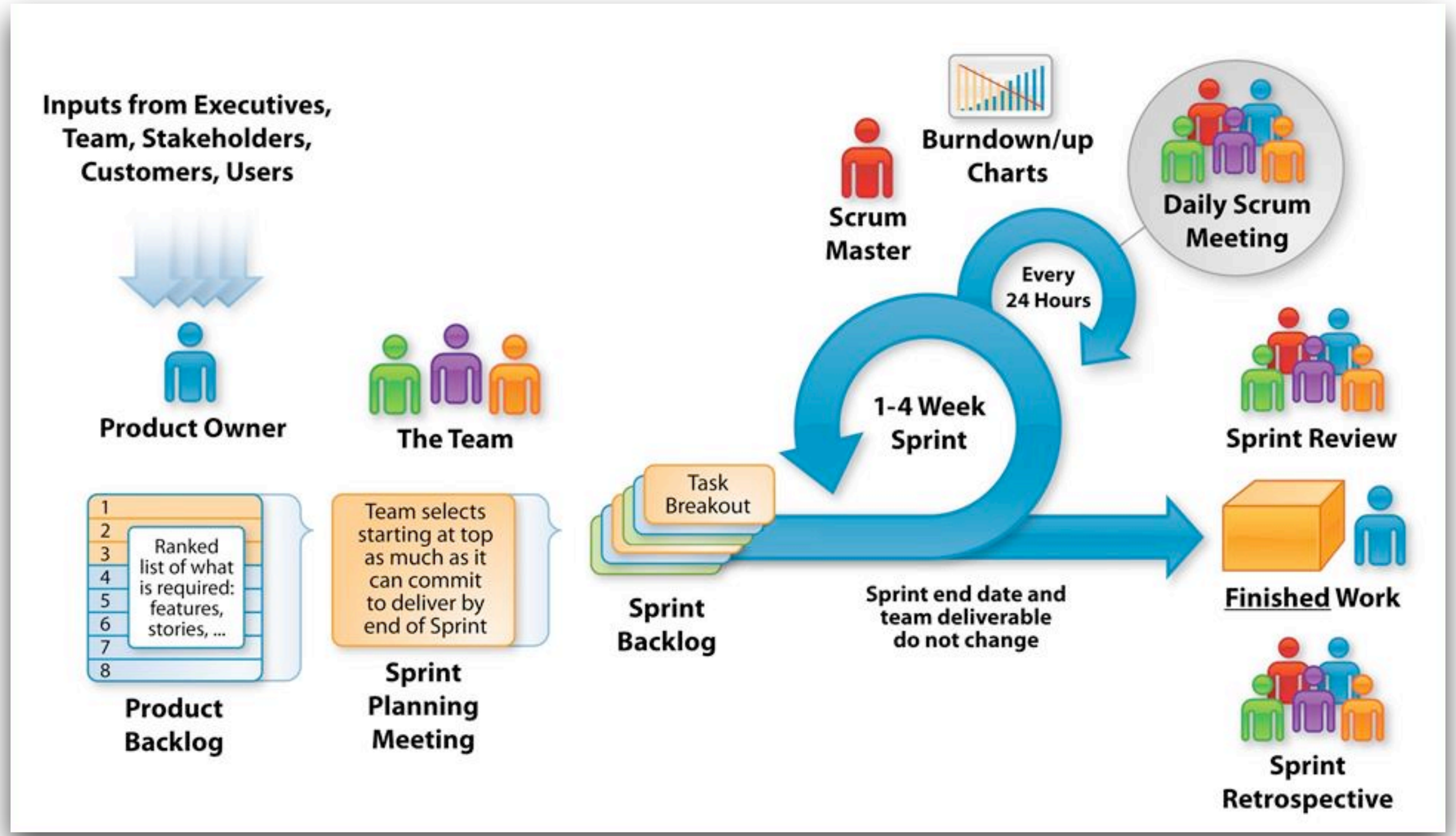
Your role:

- You should aim to complete the project work in 3 sprints
- Spend some time (1 week) planning out your approach to the project
- Conduct a sprint planning meeting before each sprint and select the tasks from the backlog
- Conduct a sprint review (and retrospective) after the sprint
- For each sprint you should select the ScrumMaster (rotate this role among all team members).
- Hold daily standups (organised by the Scrum Master)
- Use tools (eg. Trello boards)

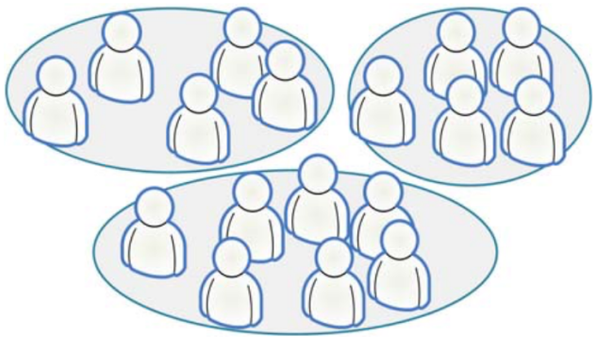
Scrum

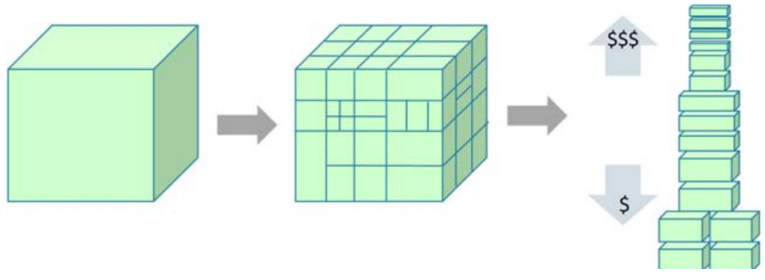

Scrum Sprint Cycle
1-4 weeks
Product Backlog
Feature A
Feature B
Feature C
Sprint Planning
Sprint Backlog
Daily Scrum
Sprint Execution
Potentially Shippable Product Increment
INSPECT
ADAPT
Sprint Retrospective
INSPECT
ADAPT
Sprint Review

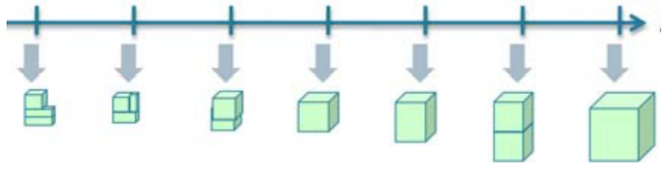
Scrum


Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Sprint Retrospective

Scrum







Split the organisation into teams

Split the work into concrete deliverables

Split time into short fixed-length iterations, with potentially shippable code demonstrated after each iteration.

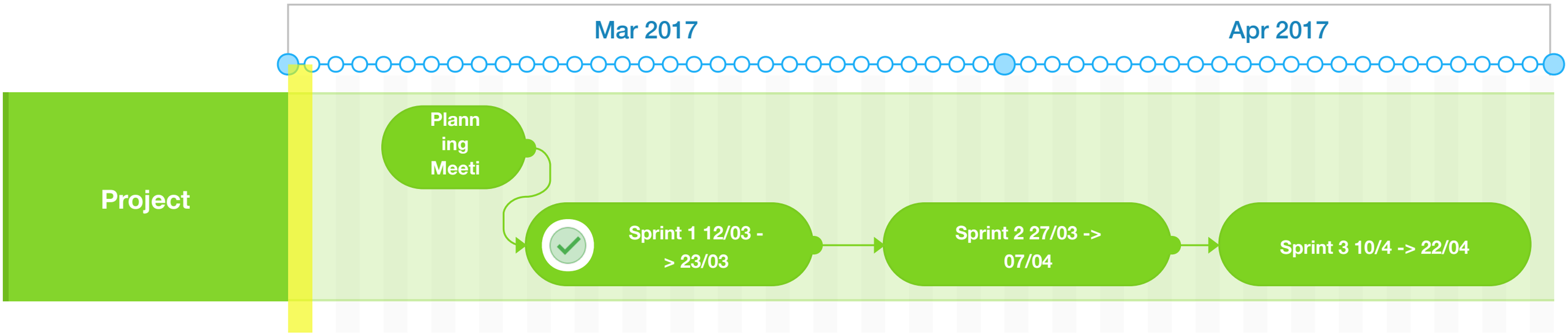
Optimise the release plan and update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration.

Optimise the process by having a retrospective after each iteration.

Lean principles

- Were there things you had to do that were futile, useless, or superfluous? This is muda 無駄—work you had to do that didn't add value.
- What about times when you were sitting around idle, anxiously waiting for someone to get back to you so you could get your work done? This is mura 斑—unevenness, or work that happens in fits and starts.
- Those times that you had to stay late or work weekends because you were expected to do more work than was humanly possible? That's muri 無理—overburdening, or being expected to do unreasonable or impossible things.
- Kaizen 改善 continuous improvement. By improving processes, kaizen aims to eliminate waste.

Project Plan



Before:
Sprint Planning meeting

Choose a Scrum Master

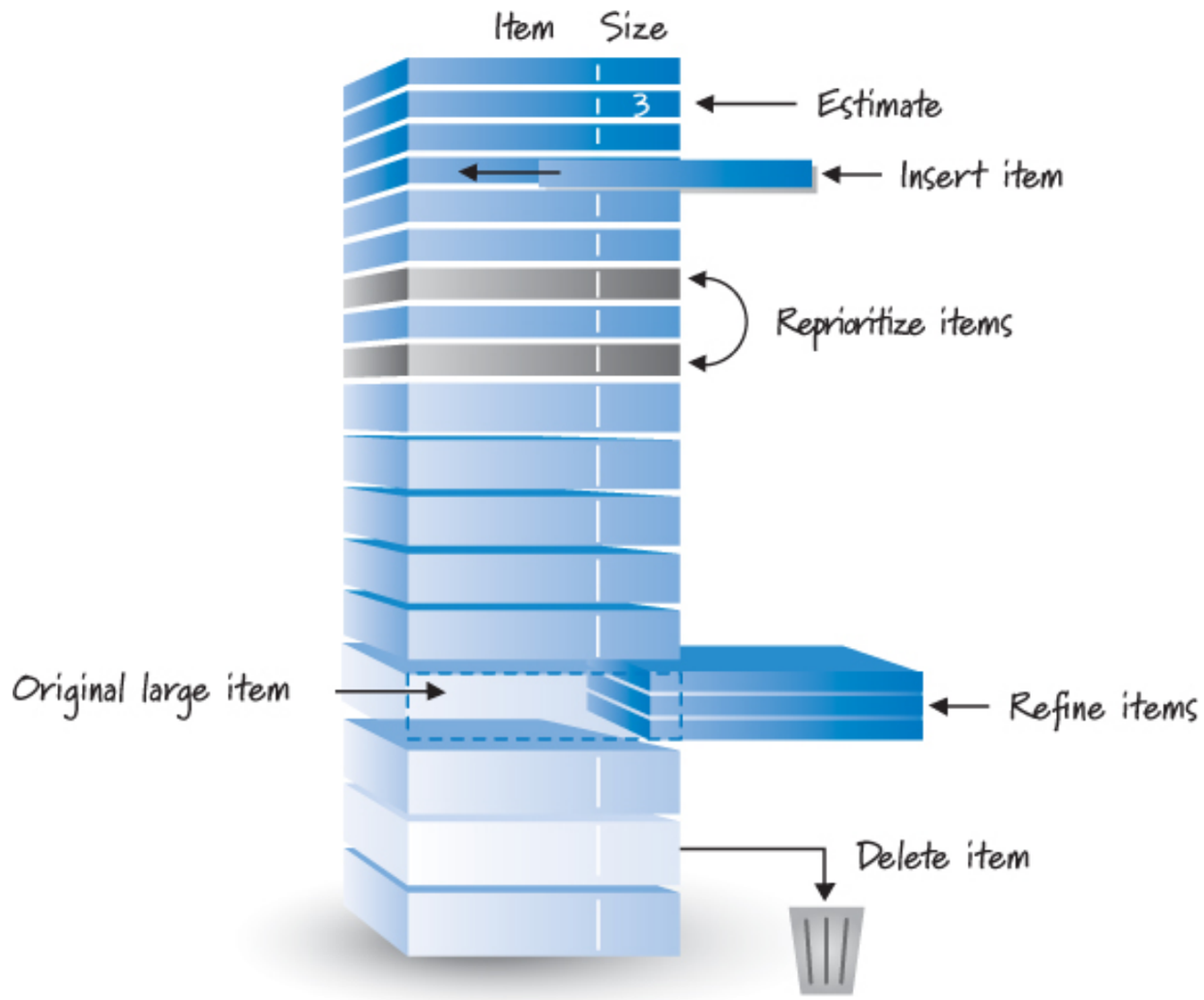
During:
Standup meetings

After:
Sprint Review
Sprint Retrospective

Sprint - Product Backlog

The Product Backlog changes with time

- Items get done and removed from the backlog
- New features are identified and added
- The team spots defects the need fixes
- Vague long term items get clarified and split into many simpler, do-able, near term items



Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-16 hours)
  - Collaboratively, not done alone by the ScrumMaster

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

Daily standup

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
- Helps avoid other unnecessary meetings

3 Questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are *not* status for the ScrumMaster
  - They are commitments in front of peers

Sprint Execution

The daily scrum: stand up team meet to sync up and plan the day. In turn, everyone answers the questions:

What did I accomplish since my last scrum?

What do I plan to work on until the next scrum?

Am I on schedule?

Are there, or do I foresee, any problems?

- **Do not try to solve problems in the scrum.**
- **Update the Sprint Backlog accordingly.**
- **The team should work at a sustainable pace.**

Sprint Review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal, 2-hour prep time rule, no slides

What features are working well?

- What features are not working so well?
- Should the Product Backlog be changed?

• You will update the Product Backlog based on the review outcome

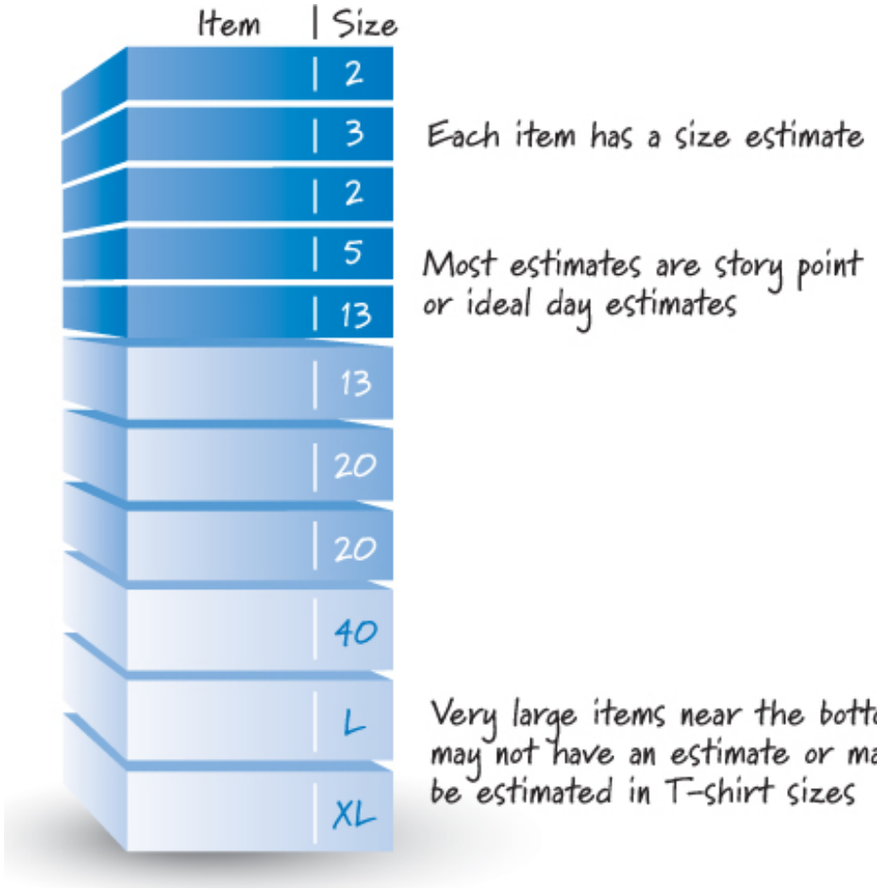
• You will sort the Product Backlog according to latest priorities

Sprint Retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - Product owner
  - Team

Product Backlog

- **The development team estimates the time that each feature will take (in days/hours).**



Project Work

The ScrumMaster should chair scrums at regular times, e.g. 2/3 times per week

- **You should share your "Sprint Notes" with your team (eg. use Google Docs)**
- **For each sprint, the Sprint Notes could contain:**
  - Sprint backlog: task, initials, estimate,
  - Test plan, if manual tests ONLY
  - Sprint review & retrospective summary
- **For each sprint, you should finalise/commit:**
  - commit source code
  - your team's Sprint Notes
  - your individual Learning Journal

Sprint Notes

The Sprint Notes are meant to be a *byproduct* of doing the sprint, not documentation that you create just before the submission deadline.

- **The Learning Journal is a personal reflection on the sprint that you *do* create after the sprint is over.**

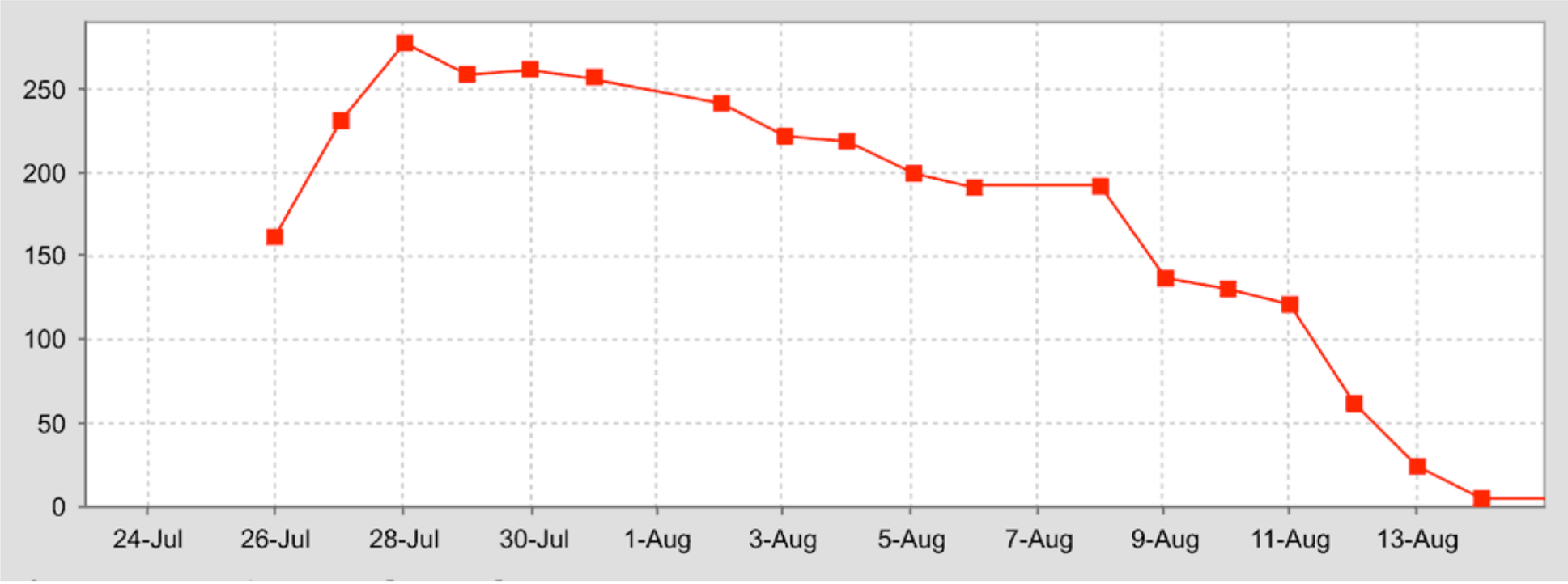
Example Backlog

Sprint Backlog Template

Backlog Item	Story Points	Responsible	Status	Original Estimate	Day 1	Day 2	Day 3	Day 4	Day 5	Sprint Review
User Story #1	**8**									
Task				7	5	3	0	0	0	0
Task				3	1	1	5	0	1	0
Task				1	0.5	0	3	0	0	0
Task				0.5	1	2	3	1	0	0
User Story #2	**1**									
Task				3	3	0.5	0.5	0	0	2
Task				3	5	5	1	1	1	0
Task				2	2	5	0	1	0	1
Task				5	5	9	5	1	0	1
User Story #3	**5**									
Task				8	6	0	0	0	0	0
Task				3	1	3	3	3	0	0
Task				1.5	1	0.5	0.5	1	1	0
Task				2	0.5	0	0	0	0	3
User Story #4	**8**									
Task				9	4	2	2	1	1	0
Task				6	6	3	3	3	1	1
Task				6	2	8	8	1	0	1
Task				0.5	0.5	0.5	0.5	0	0	0
User Story #5	**3**									
Task				2	1	1	1	0.5	1	1
Task				6	6	6	0.5	3	9	0
Task				9	9	9	4	3	3	3
Task				0.5	0.5	0.5	1	0.5	0	1
Total				78	60	59	41	20	18	14

Burndown Charts

- you need one for each sprint (x3)
- and one for the whole project (tracks overall progress)



Indicates total remaining team task hours within one Sprint
Re-estimated daily, thus may go up before going down
Intended to facilitate team self-organisation

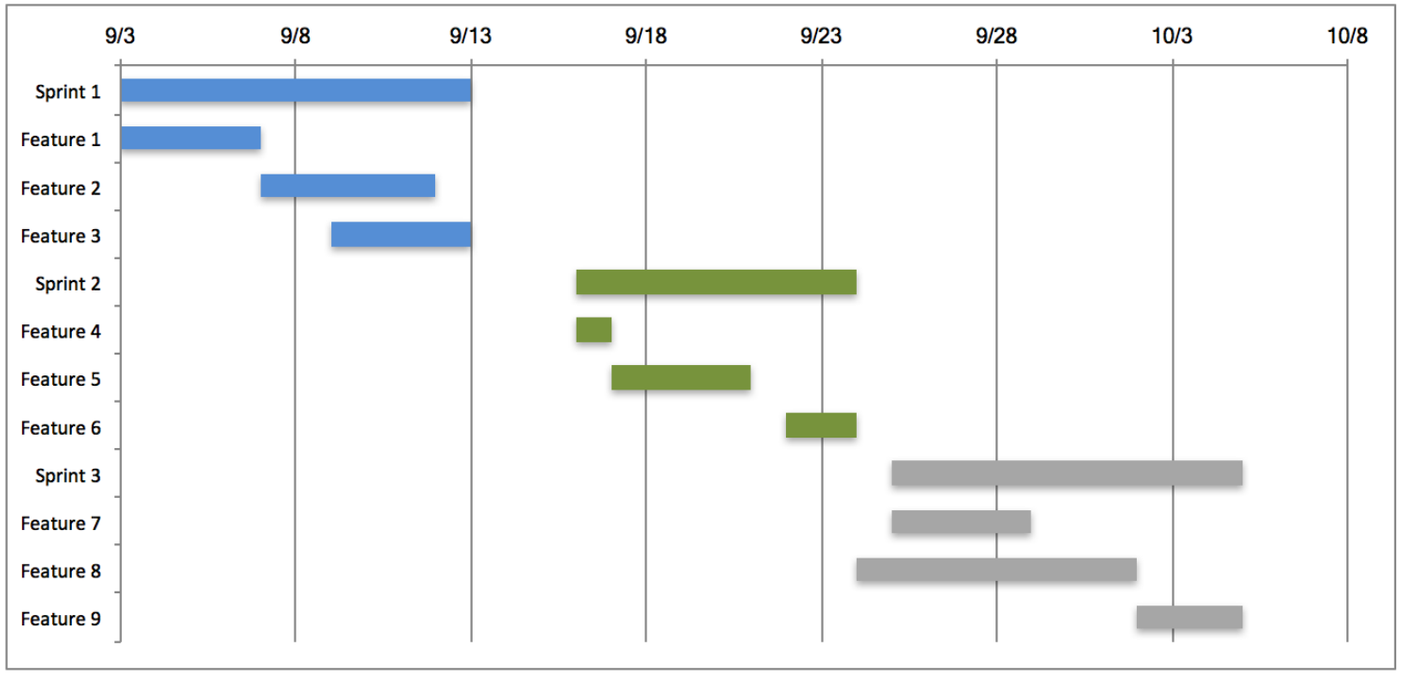
Project Plan

Agile Project Plan

Project Name Product Release
Project Manager Alex B.
Project Deliverable
Scope Statement

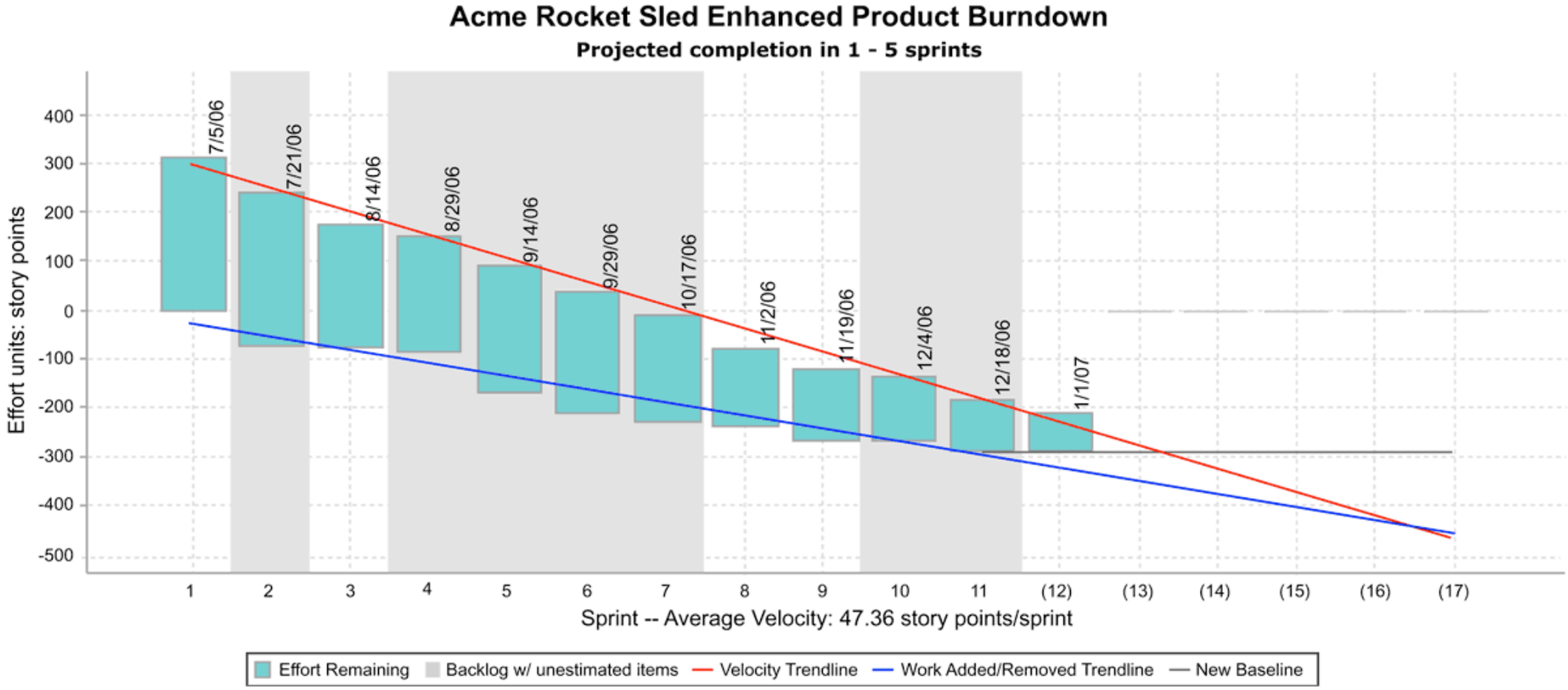
Start Date 03-Sep
End Date 05-Oct
Overall Progress 20%

Task Name	Responsible	Start	End	Days	Status
Sprint 1	Alex B.	9/3	9/13	10	Complete
Feature 1	Frank C.	9/3	9/7	4	Complete
Feature 2	Jacob S.	9/7	9/12	5	Complete
Feature 3	Jacob S.	9/9	9/13	4	Overdue
Sprint 2	Jacob S.	9/16	9/24	8	In progress
Feature 4	Alex B.	9/16	9/17	1	In progress
Feature 5	Frank C.	9/17	9/21	4	Not started
Feature 6	Shari W.	9/22	9/24	2	Not started
Sprint 3	Shari W.	9/25	10/5	10	Not started
Feature 7	Alex B.	9/25	9/29	4	Not started
Feature 8	Kennedy K.	9/24	10/2	8	Not started
Feature 9	Jacob S.	10/2	10/5	3	Not started



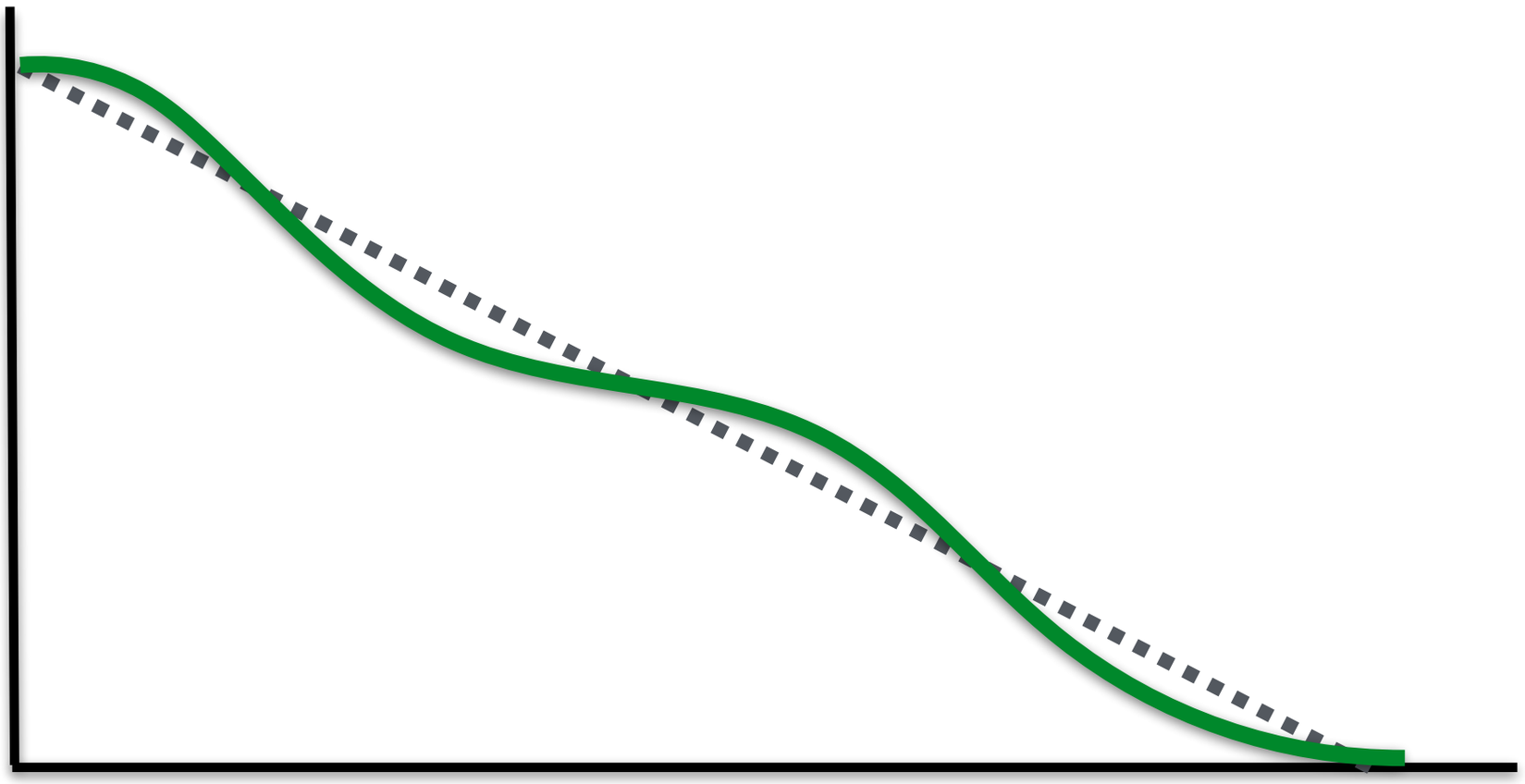
Burndown Charts

- you need one for each sprint (x3)
- and one for the whole project (tracks overall progress)



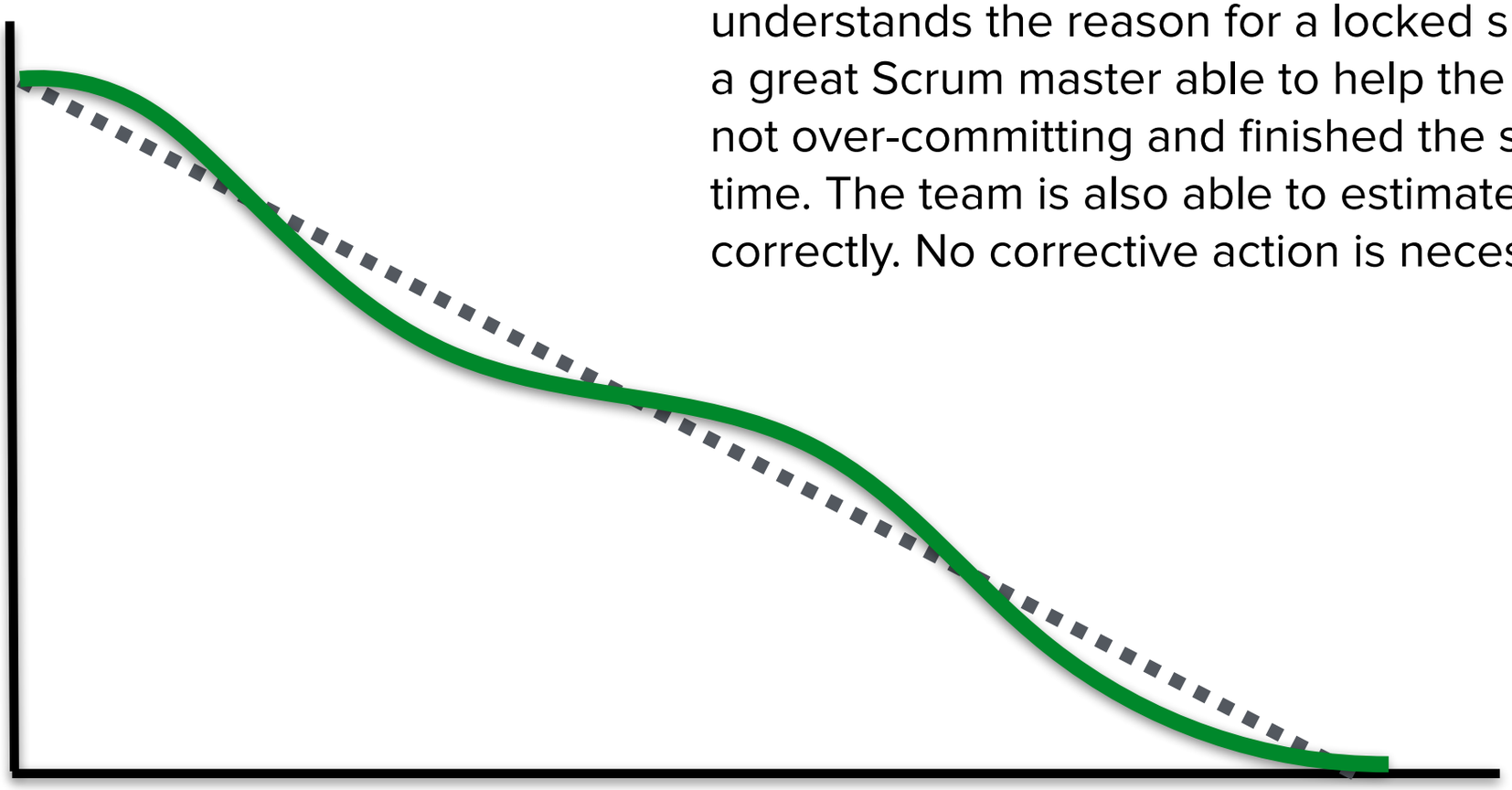
Burndown charts

Burndown charts



Time unit (hours or days) represents remaining time necessary for completion.

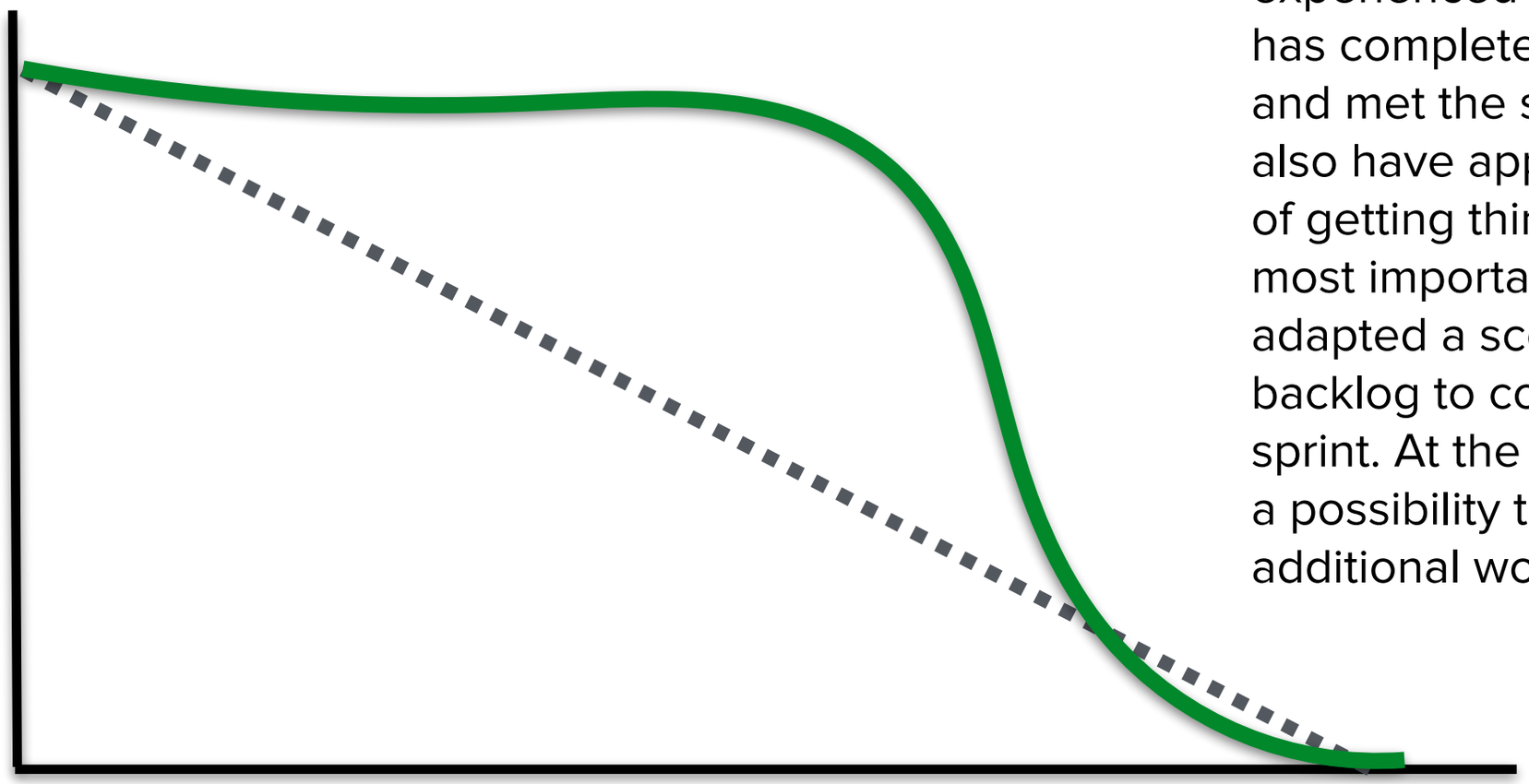
Burndown charts



Such diagram indicates the great team able to organise itself. It indicates a great product owner who understands the reason for a locked sprint backlog and a great Scrum master able to help the team. The team is not over-committing and finished the spring backlog on time. The team is also able to estimate capacity correctly. No corrective action is necessary in such case.

Time unit (hours or days) represents remaining time necessary for completion.

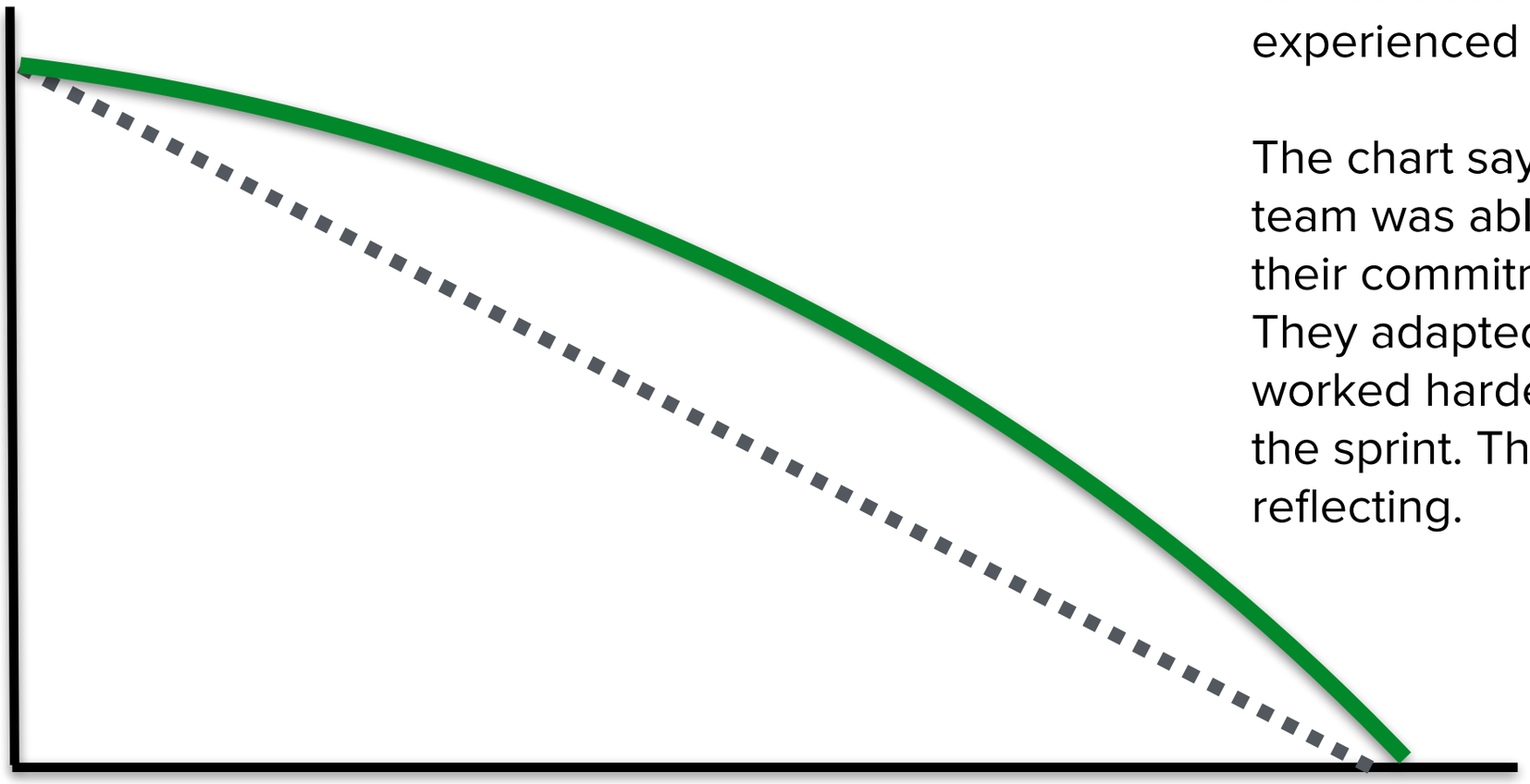
Burndown charts



Such progress might be observed on charts of experienced teams. The team has completed work on time and met the sprint goal. They also have applied the principle of getting things done, but the most important is they have adapted a scope of the sprint backlog to complete the sprint. At the end the team has a possibility to complete some additional work.

Time unit (hours or days) represents remaining time necessary for completion.

Burndown charts

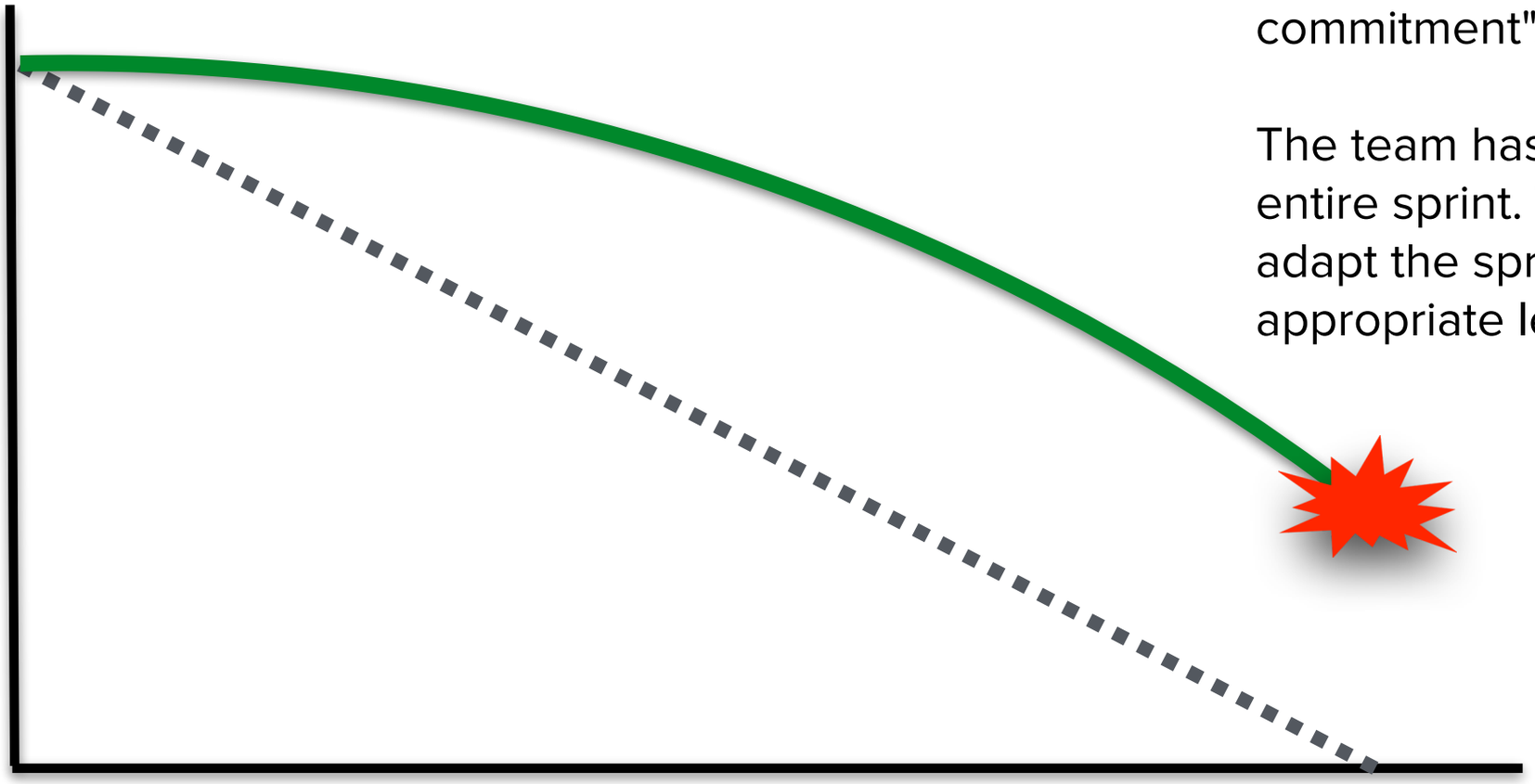


This is a typical progress that can be observed in many experienced agile teams.

The chart says again that the team was able to complete their commitment on time. They adapted the scope or worked harder to complete the sprint. The team is self-reflecting.

Time unit (hours or days) represents remaining time necessary for completion.

Burndown charts

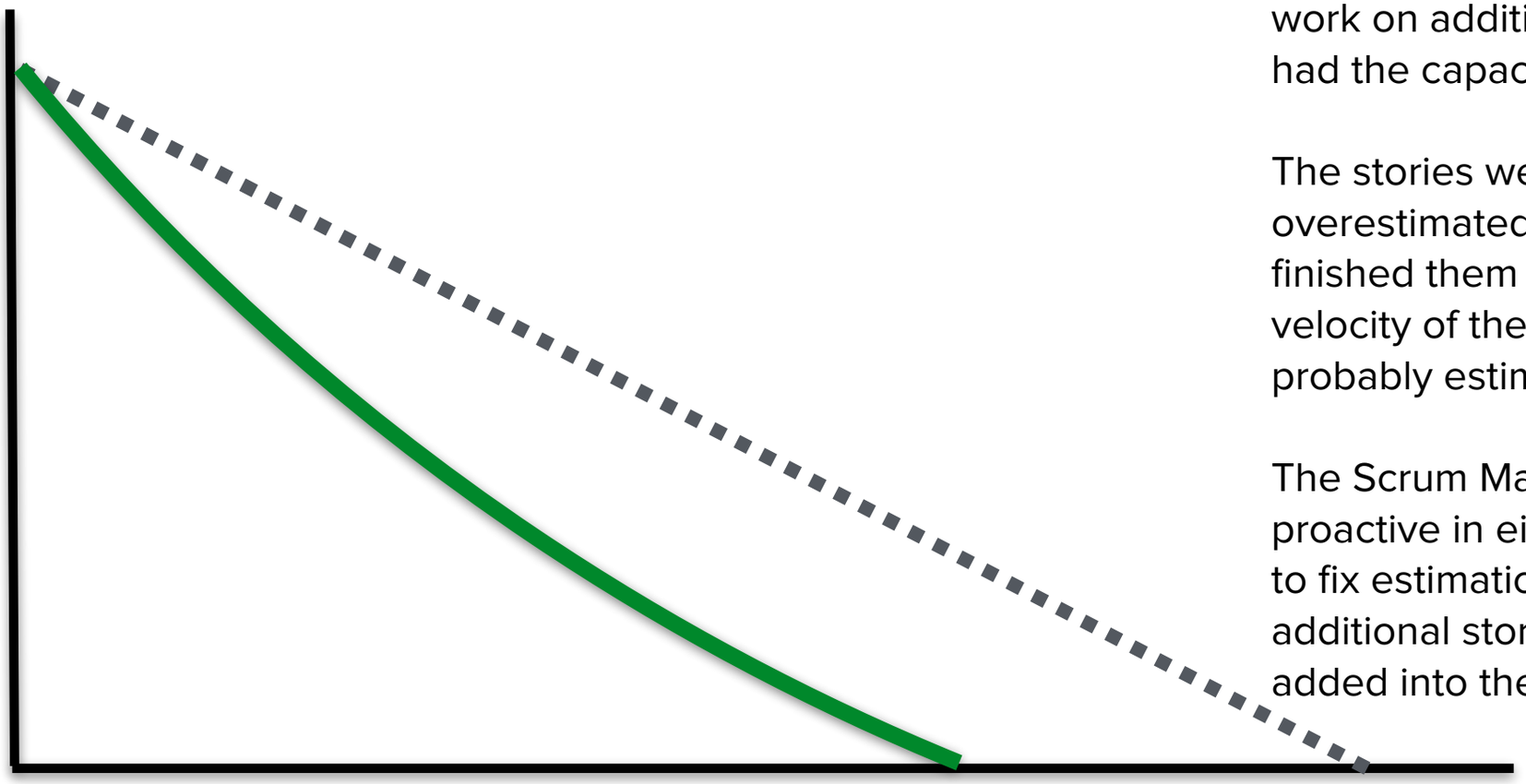


This burndown chart says: "You have not completed your commitment".

The team has been late for the entire sprint. The team did not adapt the sprint scope to appropriate level.

Time unit (hours or days) represents remaining time necessary for completion.

Burndown charts



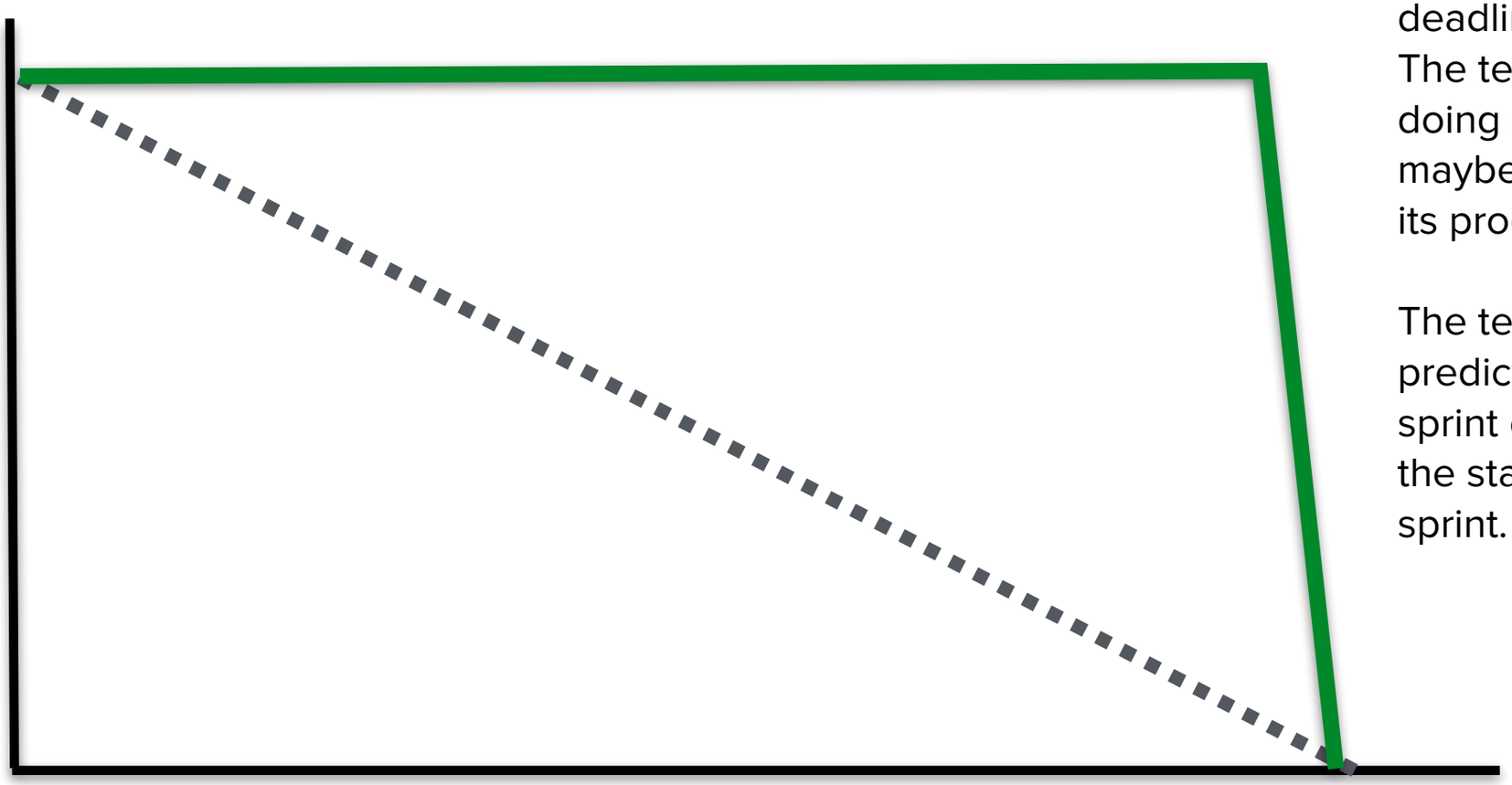
The team finishes its work sooner than expected. The stories were implemented, but the team didn't work on additional stories even it had the capacity to do it.

The stories were probably overestimated, therefore the team finished them earlier. Also the velocity of the team has not been probably estimated correctly.

The Scrum Master must be more proactive in either getting the team to fix estimation or ensure additional stories are ready to be added into the current sprint.

Time unit (hours or days) represents remaining time necessary for completion.

Burndown charts



OOPS! There's a deadline!
The team is probably doing some work, but maybe it does not update its progress accordingly.

The team is not able to predict the end of the sprint or even to provide the status of the current sprint.

Time unit (hours or days) represents remaining time necessary for completion.

Learning Experience

Rotate the type of tasks that you do for each assignment so that everyone does a bit of everything

- **Rotate the ScrumMaster role between assignments**
- **Allocate the tasks evenly**
- **If someone is finding a task difficult, help them do it, don't take it from them**
  - This is a two-way street, if you help them do it for the first assignment, you won't have to do it yourself for the rest of the assignments (enlightened self-interest)
- **Within a team should be a non-competitive environment**
- **Usually the best functioning teams produce the best software, not the "lone rangers".**

Example Learning Journal

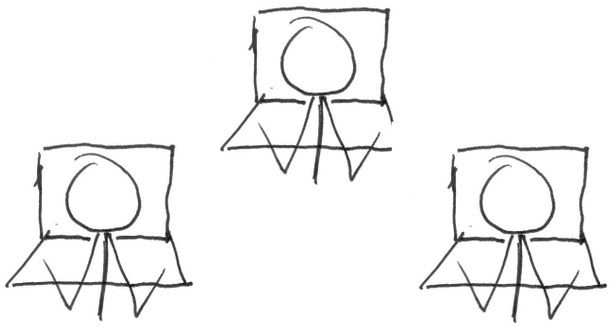
Percentage of Group Submission

- State the percentage of the group submission that was done by you (include coding and testing). Ideally, 33%.

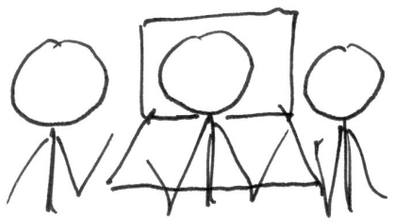
- **Skills Practised**
  - Bullet points describing skills that you exercised while doing the assignment.
- **Learning Done**
  - Bullet points describing what you learned while doing the assignment.
- **Reaction**
  - Bullet points describing how you felt about the experience of doing the assignment.
- **Goals**
  - Bullet points describing what you plan to practise and/or learn in the next assignment.

Teamwork

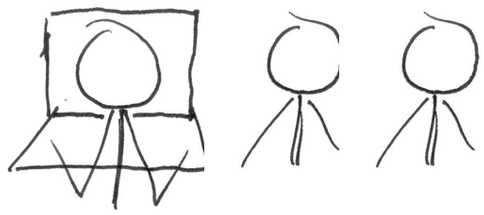
Types of Teams



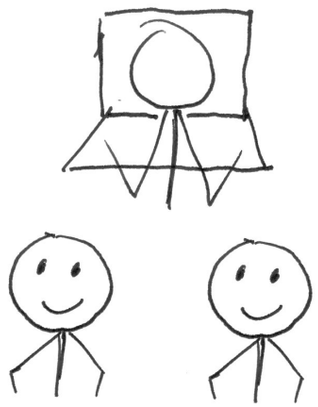
Divide and conquer



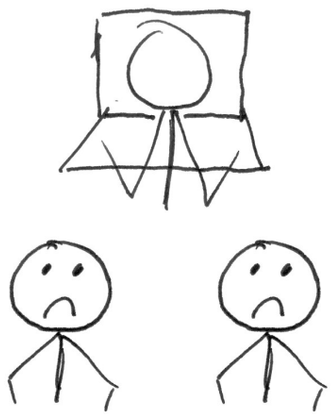
trio



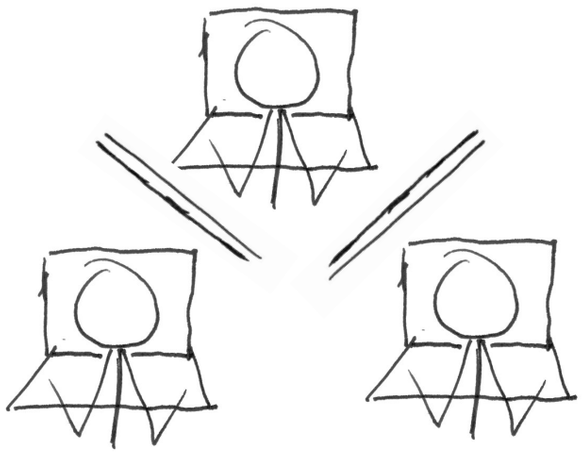
relay



drop outs



one man band



every man for himself

Characteristics of Good Teams

- How to achieve these characteristics?
- What habits do you need?

Characteristics of Good Teams

- **Commitment**
  - Work hard: Don’t expect others to do what you won’t do yourself
  - Want the project to succeed
- **Participation**
  - Include everyone in discussions
  - Ask people’s opinion
  - Don’t be shy
  - Understand why it is being done this way
  - Don’t zone out
  - Who is doing what? Clearly assign tasks and/or roles. Rotate them.

Characteristics of Good Teams

Communication

Regular discussions - meeting, email, skype, call, messaging

Explain why things were done a certain way

Discuss, don't argue

Keep to the point

Give you opinion, provided it is constructively

Be responsive, e.g. "Can't do it now, will have it done by 9."

Respect

Be on time

Listen to others and consider their opinion

Trust

Do what you said you will do, when you said you will do it.

Characteristics of Good Teams

Support

- Help each other
- Offer to help
- Allow space for each other to work

Effective decision making

- Be aware of when a decision is needed (and when it isn't)
- Sometimes a poor decision is better than no decision
- Have reasons for your decisions
- Write down your decisions in an email

Fun

- Meetings over coffee
- Have a laugh
- Rewards when something if finished or working

Meetings

What is the objective of the meeting (goals)?

- **What topics do you need to talk about (agenda)?**
- **How long will each item take?**
- **What preparation is needed?**
- **When & where**
- **Hold the meeting**
- **Write down the findings (during the meeting)**
- **Write down the action points (during the meeting)**
- **Review the agenda & objectives, is everything covered?**
- **Email the minutes**

Example Meeting log

- **keep meeting logs in Google Docs or git**

```
Monday 19 Feb 10:00-11:00
CB, AbH, AhH
WiFiLoc8 Weekly Status Update

Latest results from Spotlight:
1AP, mean error 4m, random position, random orientation

Discussed ideas for TOA:
* Discriminant Analysis of xcorr output
* TDoA versus DOE signature
* Multipath reduction with beamsteering
* Impulse response beamsteered versus null. What can we extract?

Discuss SMILE papers (see AhM email)

AP CB send SMILE papers to Joe: can we do this with WARP?
AP AhH lit survey 1st draft to CB next Tues
AP AbH continue Spotlight measurements
```

Disagreements

Kick for touch

- Cool off
- Think through your and the other person's point of view
- Circle back with a facilitator
- Disentangle the argument

- state the points of agreement
- discuss the points of disagreement
- get to the core reasons of why

- **you can start work!**

Plagiarism Policy:

https://csiweb.ucd.ie/files/csi-plagiarism-policy_august2015.pdf

There is a lot of stuff online, and you must be careful that what you submit is your own work