IS52018C Software Projects

Requirements

Planning

What are

Requirements?

- What we want the software **to do**.
- How we want the software **to perform**.
- A set of specific, clear, unambiguous statements.

- For a website...
- “Every page should download in less than 5 seconds.”
- “It should appeal to teenage girls.”

What are Requirements?

- Requirements are based on a set of **needs**.
- What do the **users** need?
- What do the other people involved need?
- What do other interacting systems need?
- Often a compromise between competing needs.

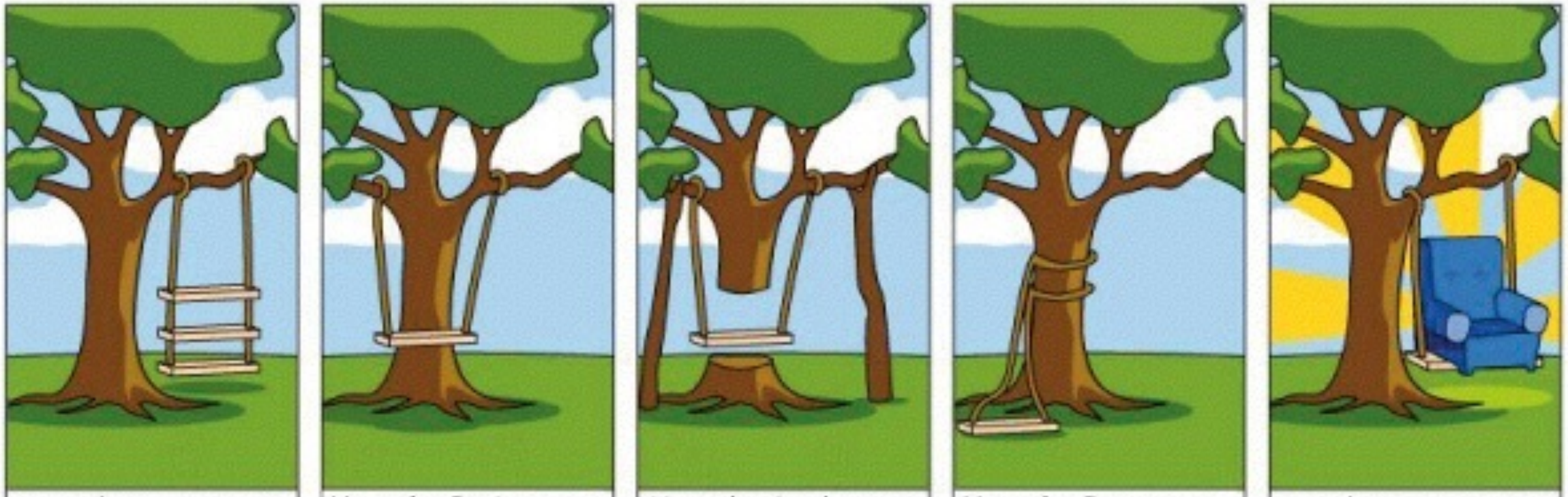
How do we establish

Requirements?

- Gather/analysis data on relevant **needs**
- Turn this into a set of **draft** requirements
- Iterate the requirements, until they are...
- Practical and feasible (we can do this)
- Specific, clear, unambiguous (we'll know when we're done)
- And **stable** (not going to change)

Why bother with Requirements?

- Hmm... this sounds like hard work. Can't we just start writing the software?
- No requirements = no target to aim for.
- If you manage to build anything, it won't be what was really needed.
- Not getting the requirements right is a major cause of real world project failure.


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Kinds of Requirement

Functional vs Non-Functional

- **Functional**: what the system should do.
  - "Allow text to be formatted in 20 typefaces."
  - "Save documents to cloud services."
- **Non-functional:** constraints on the system.
  - "Run on PCs and Macs."
  - "Delivered in six months time."

Many different kinds of non-functional requirement

Non-Functional Requirements

Data

- What kind of data will the system use?
- Type, volatility, size/amount, persistence, accuracy, ...
- *Share dealing*: accurate, up-to-date, volatile
- *Personal banking*: accurate, long-term, high volume

Non-Functional Requirements

Environmental

Context of use (where system will operate)

- **Physical**: dust, noise, vibration, light, heat, humidity, ...
- **Social**: collaboration, coordination, sharing, syncing, separation, privacy, trust, ...
- **Organisational**: hierarchy, attitude, remit, support, training, resources, communications, ...
- **Technical**: platforms, limitations, ...

Non-Functional Requirements

User Characteristics

- Who are the users? Abilities, habits & attitudes.
- Basic human physiology, psychology
- Special groups: children, visually impaired, ...
- Knowledge, skills, expertise, language, education, profession, identity, culture, religion
- **Tasks** they will carry out, frequent vs casual
- Define user profiles, often expressed as **personas**

Non-Functional Requirements

Usability Goals

- Ensure interactions meet **measurable** criteria.
- **Effectiveness**: how well the system does the job
- **Efficiency**: how productive users are
- **Safety**: the capacity for error and recovery
- **Utility**: how well it allows users to do things
- **Learnability**: how easy it is to learn to use
- **Memorability**: how well users remember to use it

Non-Functional Requirements

User Experience

Does the system need to be...?

- Fun
- Enjoyable
- Pleasurable
- Aesthetically pleasing
- Motivating

Underwater PCs

High-level Requirements

- Allow divers to record information about underwater environment.
- Work up to specific depth underwater.
- User can operate with one hand while swimming underwater in diving gear.
- Enter a number of commands: take photo, record locations etc.

Underwater PCs

Understand Your Users

- Designers/programmers often think they know more about the users than they do.
- “Of course I know what kind of users they are.”
- “I have been to school, so I know how to design for school students.”
- “If I have a good idea, certainly the users will also like it!”
- “Anyway, these users don’t know what they want.”

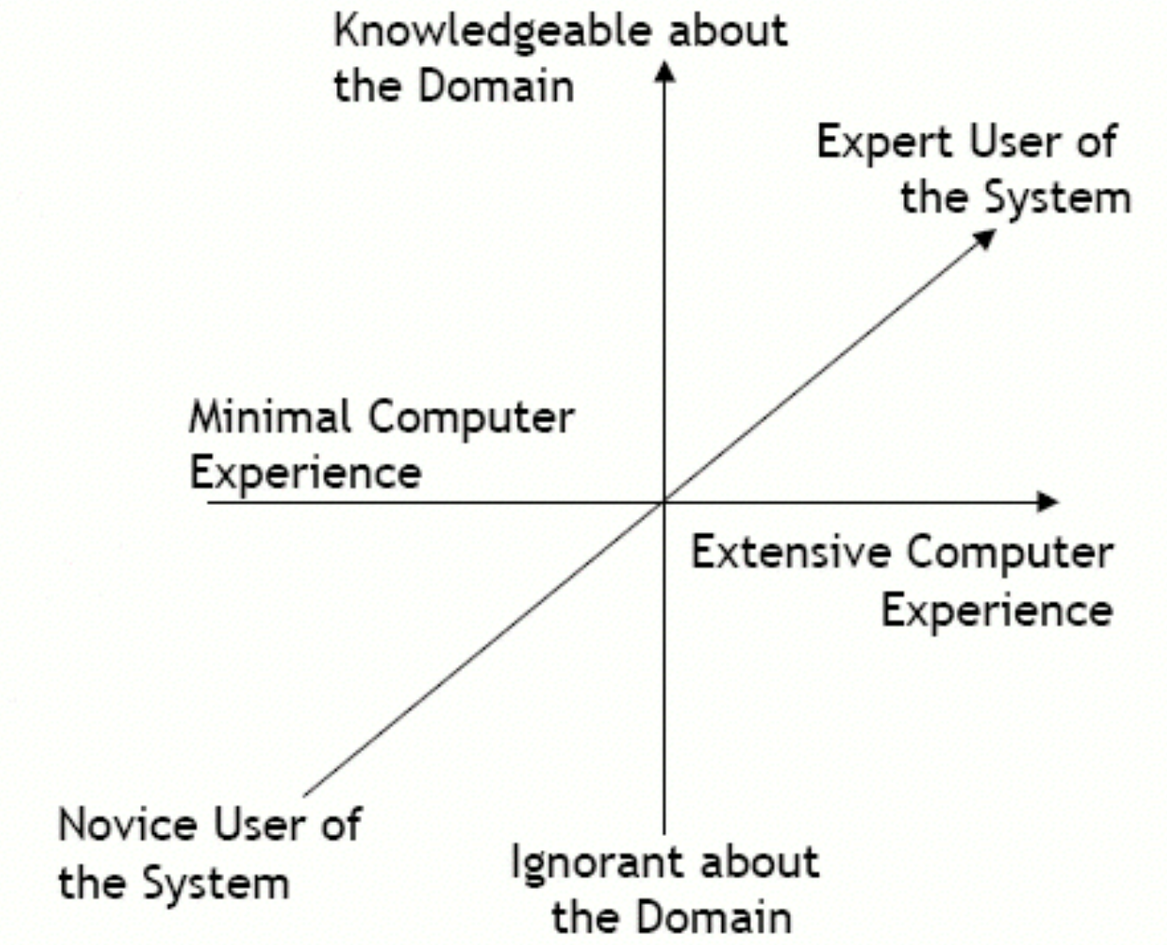
User Expertise

Different kinds of expertise:

- With this system
- With computers
- With this task

Novices: prompted, constrained, clear

Expert: flexible, powerful



Source: Jakob Nielsen, Usability Engineering, Academic Press, 1993

User Capabilities

Designing a product for children

- Require little strength to operate, more to change batteries.
- Input with reduced motor abilities
- Size of hand affect size/position of buttons
- Input/output at appropriate height
- Appropriate language for age group

Other considerations for older users or users with disabilities

User Profiles

- Write down a profile, including as much detail as you can: age, gender, physical ability, expertise, language, environment etc.
- Don’t say “this is for everyone”.
- Is there only one type of user?
- If you have identified specific users, you can begin to understand their specific **needs**.

User Needs

- It's not about asking people "what do you need?"
- We have to understand
  - Their characteristics and capabilities.
  - What they're trying to achieve.
  - How they achieve it currently.
  - Whether it would be better if they were supported differently.

Personas

- A **user archetype** representing one user group who will use the system
- A tool for thinking about design
- Guides decisions about features, interactions, and visual design
- Specific biographic details, stories

See Intro to Digital Media from last year

Timothy Powell

P. Eng, Civil Engineer
GeoLine Engineering
Age: 52

"Speed trumps security when it comes to exchanging documents. It's not worth jumping through hoops to protect a document that nobody's interested in but me and the client."

- Sends 12 documents/week at nearly 100 MB each **via FTP**
- Sends 8 documents/week under 5 MB each **via email**
- Receives 15 documents/week under 5 MB each **via email**
- Receives 15 hand-edited CAD drawings/week **via fax**
- Exchanges primarily PDF and Microsoft Word files
- Employs couriers only for

Goal: Get everything done before heading home. Timothy has a lot of work to stay on top of and firm deadlines that cannot be missed. Speed is a competitive advantage for GeoLine, so it's essential that delays do not occur. Timothy hates working at night, too, so he makes the most of his hours at the office.

Goal: Cover his back and avoid blame. In Timothy's industry, projects usually go far over budget and are completed late, at which point all the subcontractors involved begin pointing fingers at each other. Timothy needs detailed records that prove he completed exactly what was expected of him and his company.

Timothy Powell is famous among his coworkers for once visiting a construction site and remarking to the client, "Look, you may build

Stakeholders

A **stakeholder** is anyone affected by or involved in the introduction of a new system:

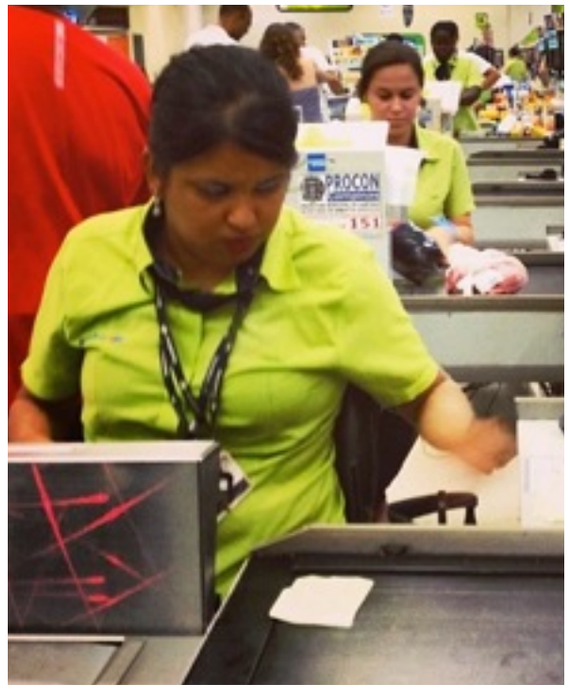
- Those who interact directly with it (users)
- Those who manage direct users.
- Those who supply input to the product.
- Those who receive output from the product.
- Those who make the decision to buy the system.
- Those who will install the system.

Requirements should reflect needs of all the relevant stakeholders, not just the users.

Who are the stakeholders?

Customers

Self-service
checkout system



Checkout staff

Managers

Suppliers

What kind of requirements would be relevant when designing…?

- Self-service filling and payment system for a petrol station.
- Clothes retail website.
- Onboard data analysis for a ship carrying geologists searching for oil.
- A motion-controlled rhythm game.

Think about…

- Functional: what the system does
- Non-functional: constraints on system, e.g. data, environment, user characteristics, usability goals, user experience.

Task Descriptions

- Descriptions of what users do now, or might do with a future system.
- **Scenarios**: an informal story, simple, natural, concrete, not generalisable.
- **Use cases**: describes a normal interaction with a *particular* system at a technical level.
- **Essential use cases**: abstract description of interaction with unspecified system.

Scenarios

It's Friday afternoon and Joe is flying to Sydney. He doesn't have enough money for a taxi to the airport, and he's running late.

He goes to the local ATM and identifies himself.

He specifies that he wants $100 from his savings account. He'd like the money in $20 notes so that he can give the taxi driver the correct change.

He doesn't want a printed receipt, as he doesn't bother keeping track of transactions in this account.

Scenarios

- No mention of bank cards or PIN numbers
- Removes focus from technical details
- Leaves open design possibilities
- Need to understand users, tasks & context
- Can be shown to users to get feedback

Shared Calendar Example

"Arrange a Meeting" Scenario

"The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals' calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people's calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in."

Shared Calendar Example

"*Arrange a Meeting*" Use Case

1. User chooses the option to arrange a meeting.
2. System prompts user for the names of attendees.
3. User enters a list of names.
4. System checks that the list is valid.
5. System prompts the user for meeting constraints.
6. User enters meeting constraints.
7. System searches the calendars for a date that satisfies the constraints.
8. System displays a list of potential dates.
9. User chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of them appointment

Shared Calendar Example

"Arrange a Meeting" Use Case

Some **alternative courses**:

5. If the list of people is invalid,

    5.1. System displays an error message.

    5.2. System returns to step 2.

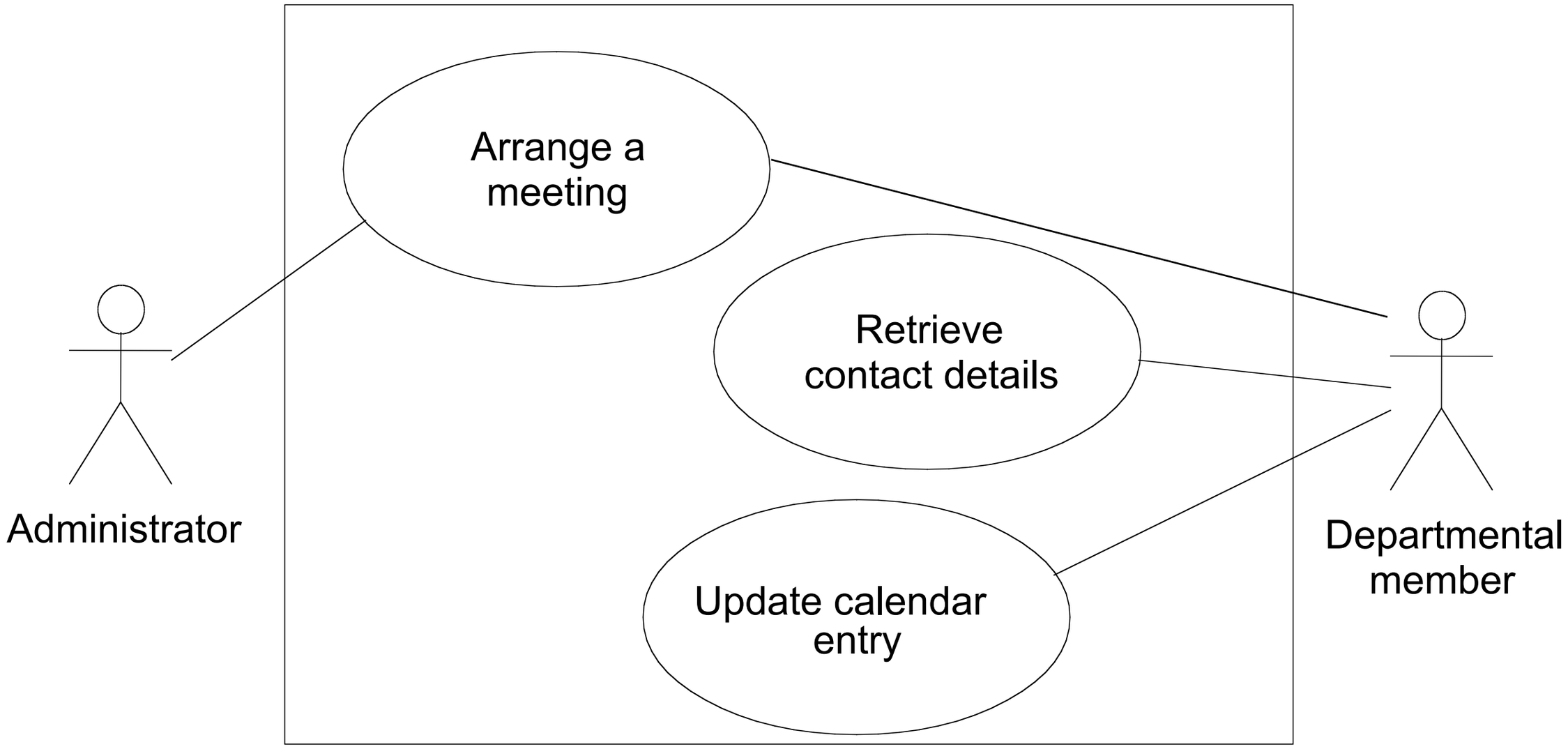
8. If no potential dates are found

    8.1. System displays a message.

    8.2. System returns to step 5.

Shared Calendar Example

A Use Case Diagram



Shows actors associated with particular use cases.

Hierarchical Task Analysis

- Task analysis is used to understand existing user-system interactions.
- Start with user goal, identify main tasks that achieve it.
- **HTA** breaks a task down into subtasks, then sub-sub-tasks and so on.
- "Plans" describe different ways of doing subtasks.
- Focus on observable actions, including those not done with software.

Hierarchical Task Analysis

Borrowing a Library Book

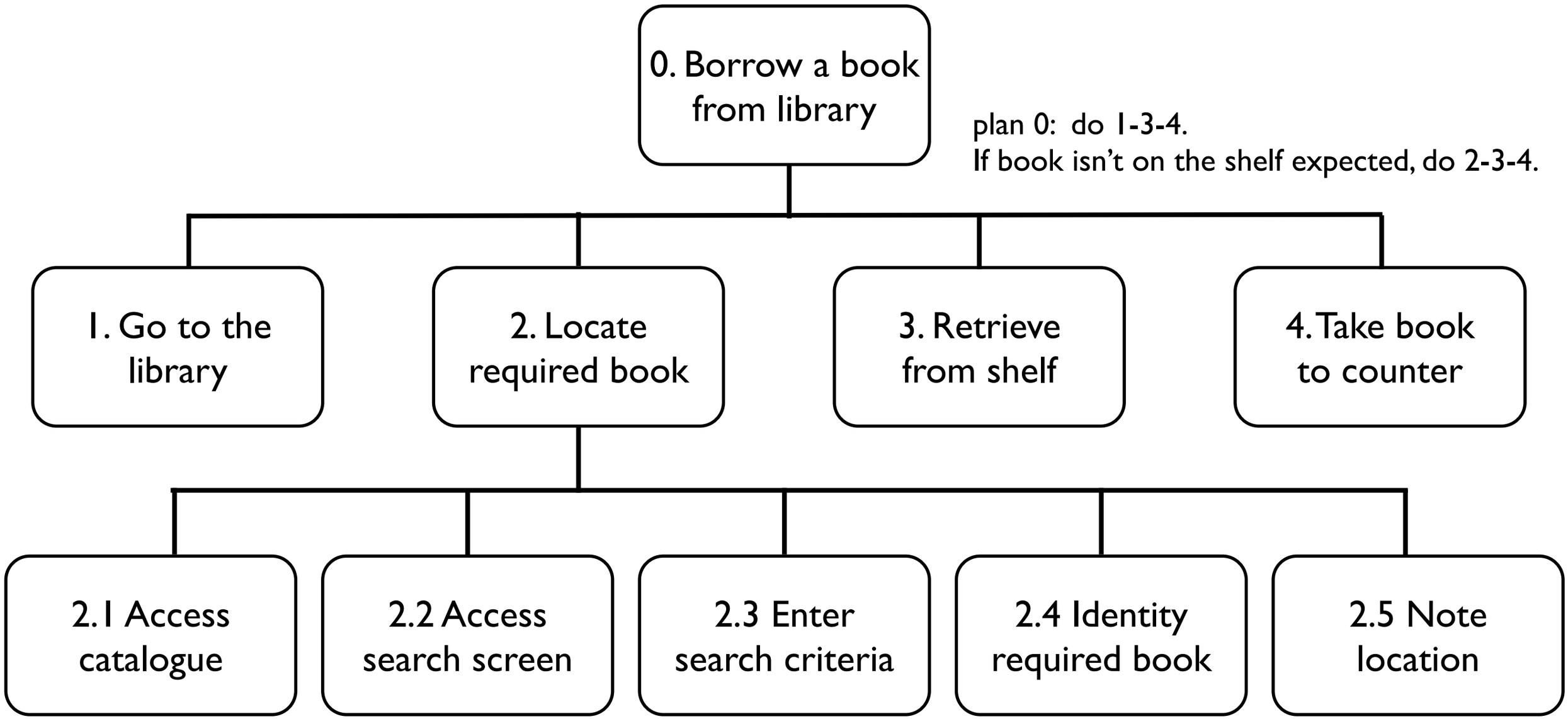
0. In order to borrow a book from the library
    1. go to the library
    2. locate the required book
        2.1 access library catalogue
        2.2 access the search screen
        2.3 enter search criteria
        2.4 identify required book
        2.5 note location
    3. go to correct shelf and retrieve book
    4. take book to checkout counter

plan at 0: do 1-3-4. If book isn't on the shelf expected, do 2-3-4.

plan at 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

Hierarchical Task Analysis

Borrowing a Library Book



plan 0: do 1-3-4.
If book isn't on the shelf expected, do 2-3-4.

plan 2: do 2.1-2.4-2.5.
If book not identified from information available, do 2.2-2.3-2.4-2.5

Project Planning

- Your proposal should outline **how** you intend to do the work.
- A plan of work
  - what needs to done.
  - when it will be done.
  - who will do it.

What to Plan?

- Aims and objectives
- Work breakdown (activities)
- Time estimates
- Sequencing
- Scheduling
- Milestones

An Example Project (Dawson 2003)

Stock Market Predictor

- Artificial neural networks (ANNs) are learning systems, modeled on the way neurons interact in the brain.
- An ANN will be trained on historical stock market data and will then be able to predict future trends.
- Aim: compare the performance of this model with more conventional systems for modeling the stock market.
- Comparison would show whether ANNs are capable of acquiring knowledge currently embodied in specialist financial software.

An Example Project (Dawson 2003)

Aims & Objectives

Aim: Develop and evaluate an artificial neural network (ANN) to predict stock market indices.

Objectives:

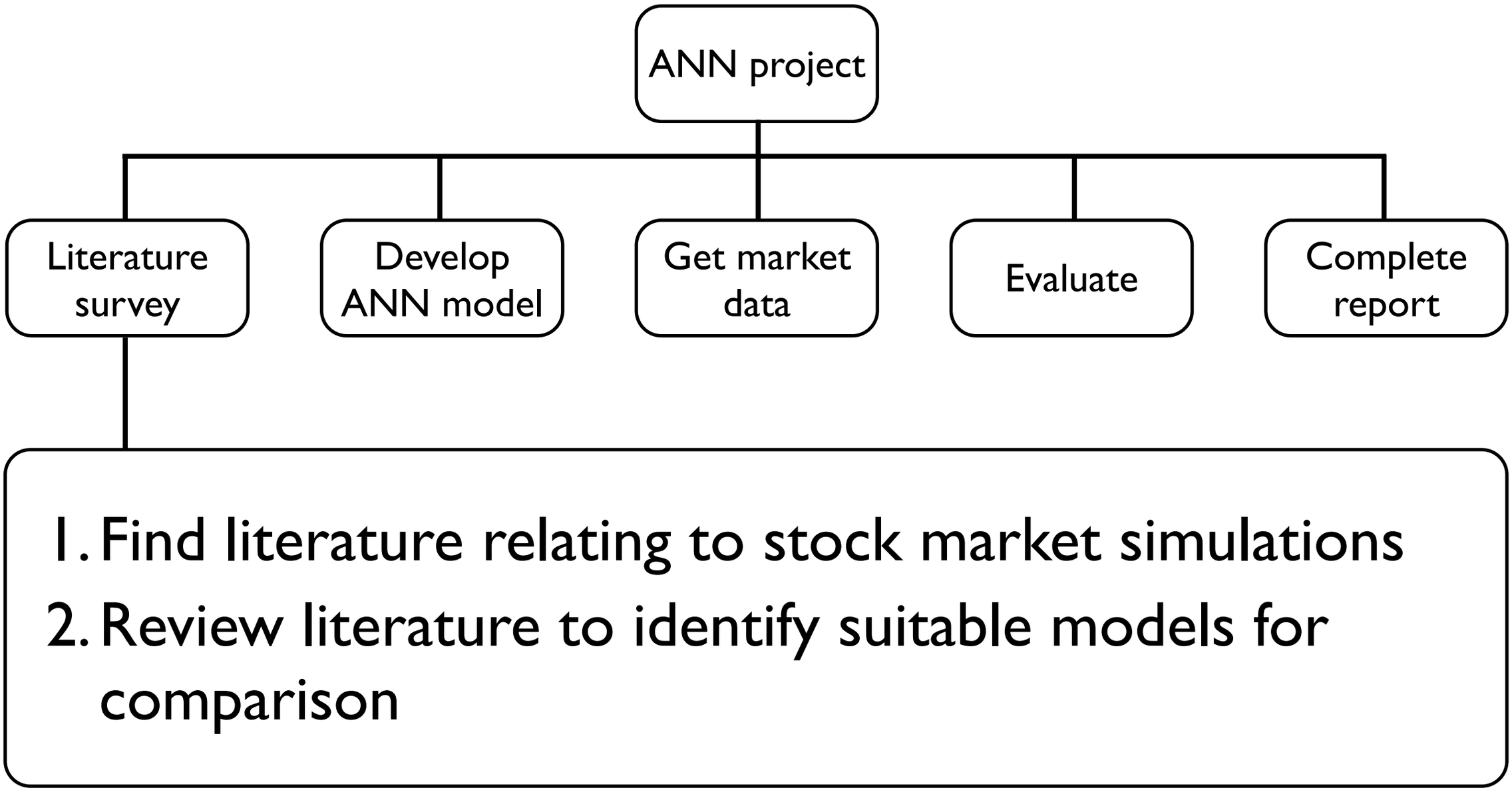
1. Review existing techniques
2. Develop suitable ANN model
3. Collect market data for analysis/evaluation
4. Compare ANN to existing techniques
5. Complete final report

Work Breakdown

- Describe main activities, e.g. one for each objective.
- Decompose this work into smaller units to identify **specific measurable** activities.
- You should be able to **estimate** the time it will take to do each activity.
- Don’t overdo it: each activity no less than 5% of work, so at most about 20 activities.

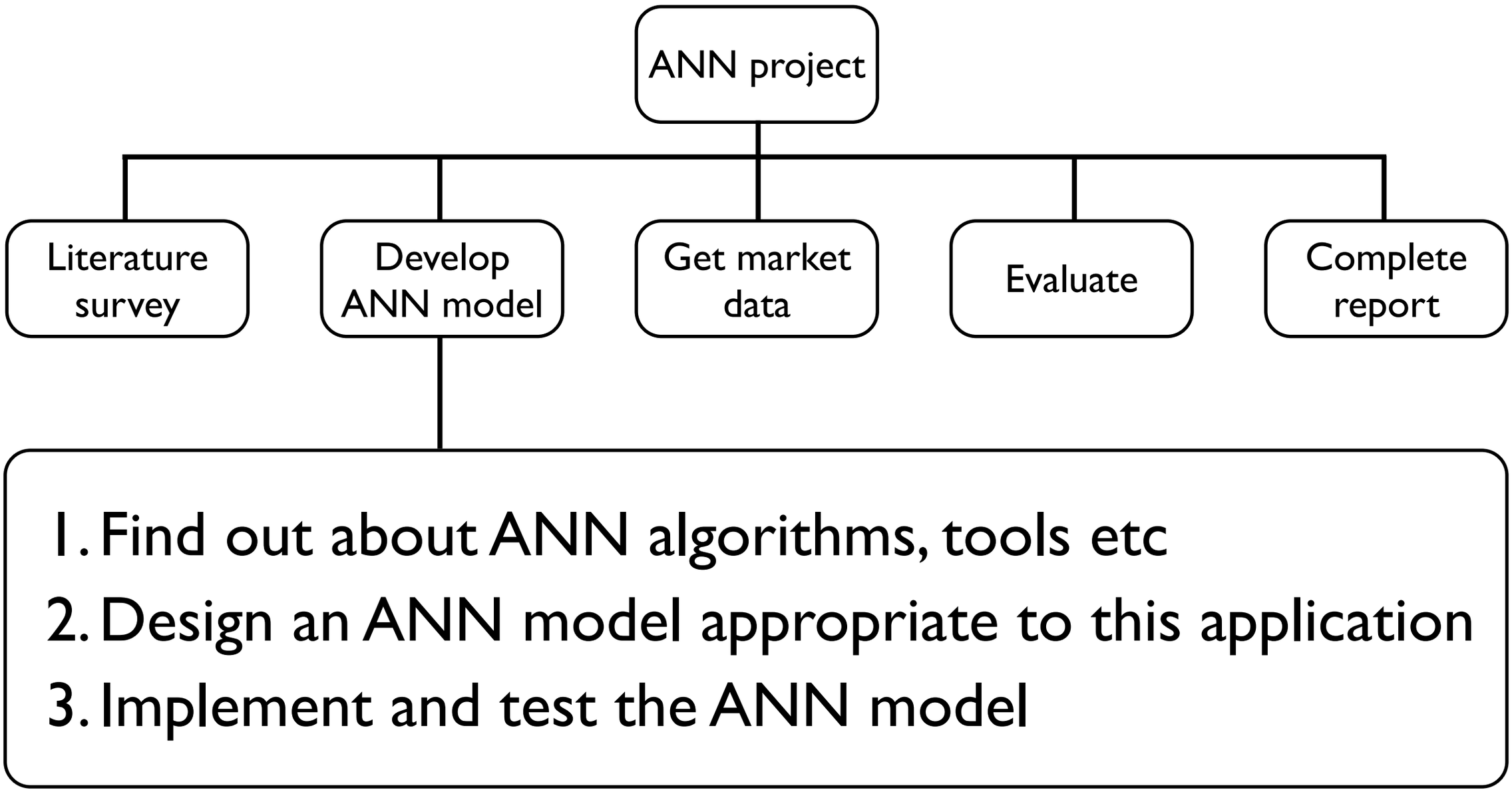
Work Breakdown

Stock Market Predictor



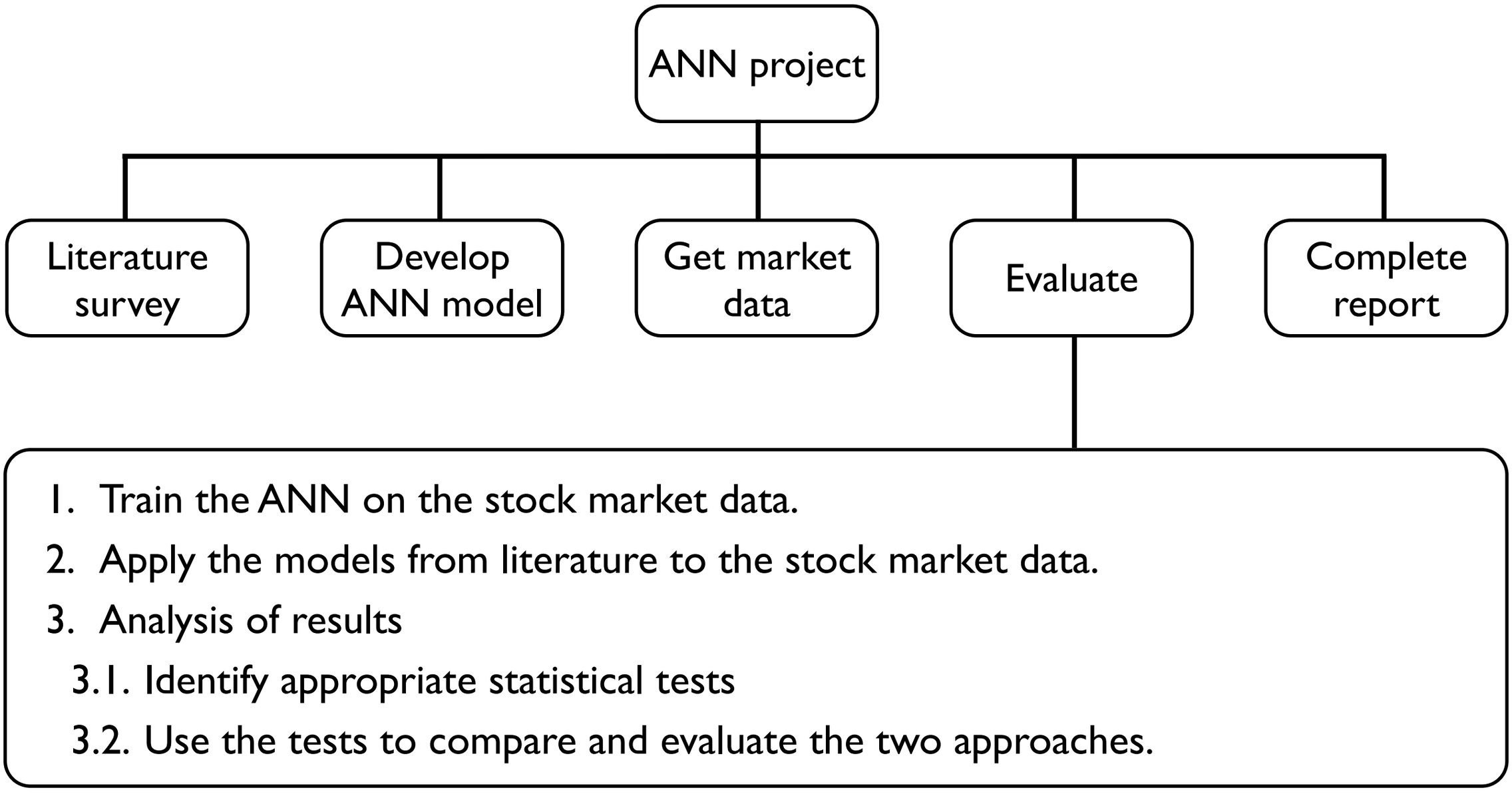
Work Breakdown

Stock Market Predictor

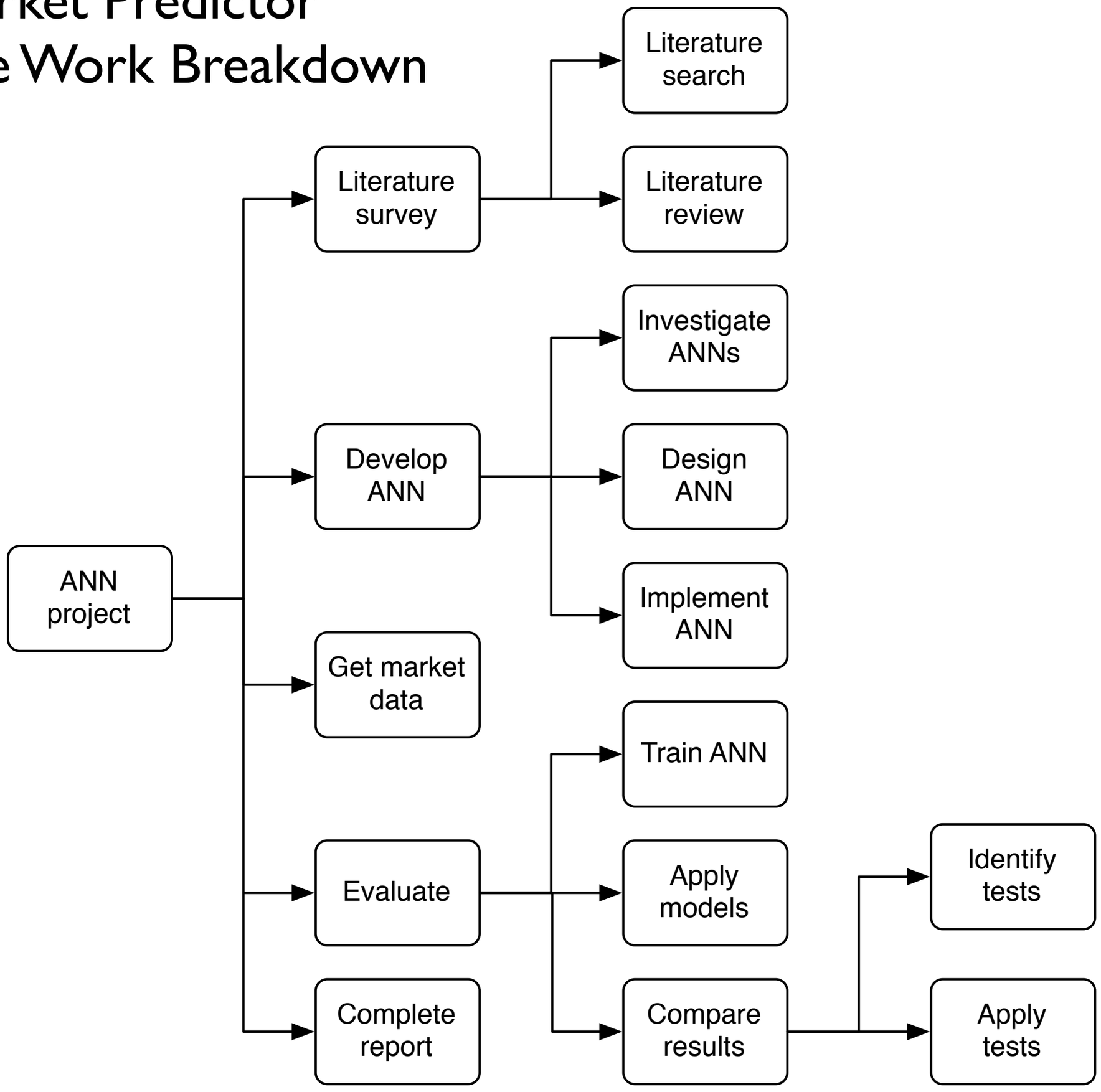


Work Breakdown

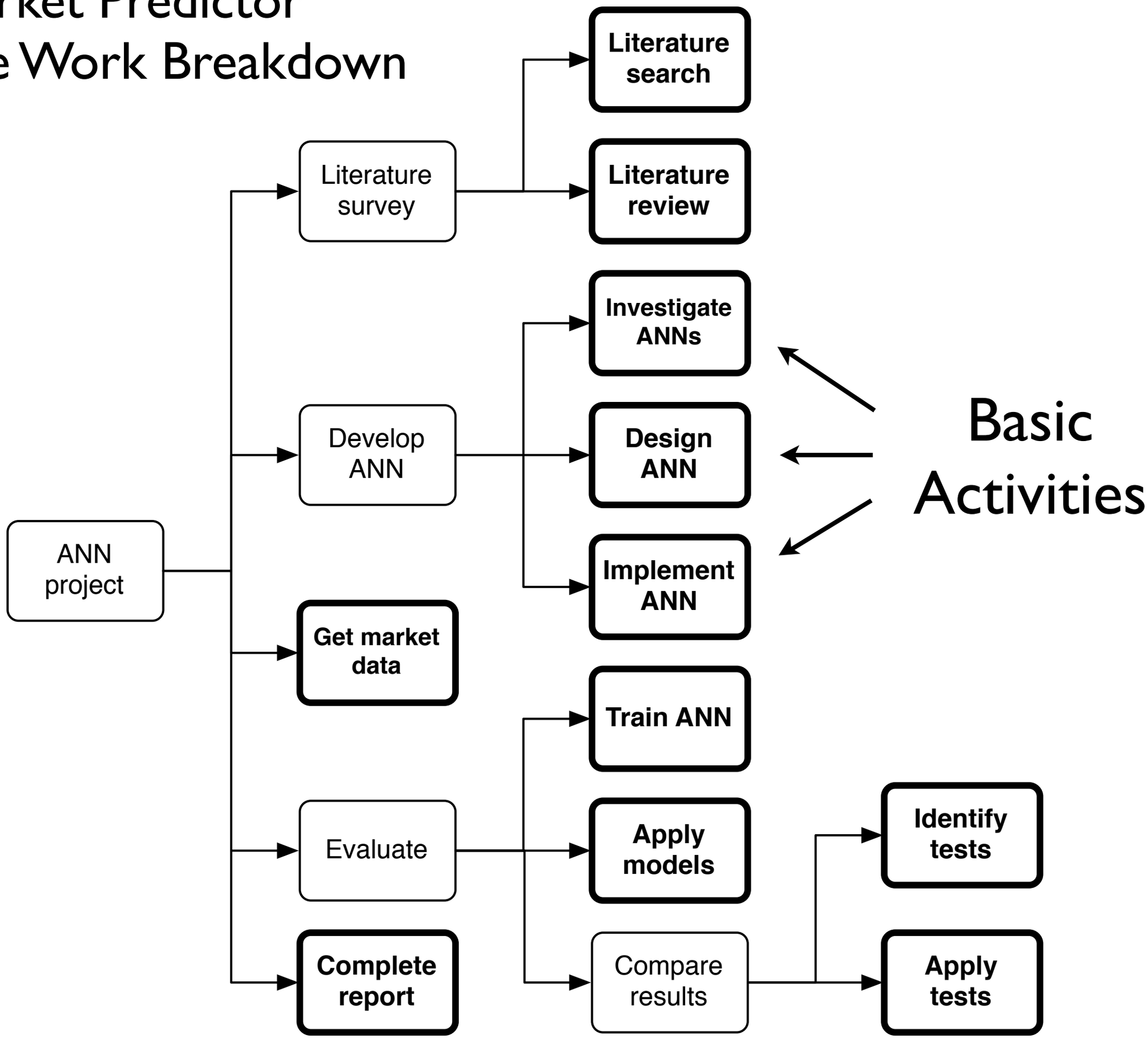
Stock Market Predictor



Stock Market Predictor
Complete Work Breakdown



Stock Market Predictor
Complete Work Breakdown



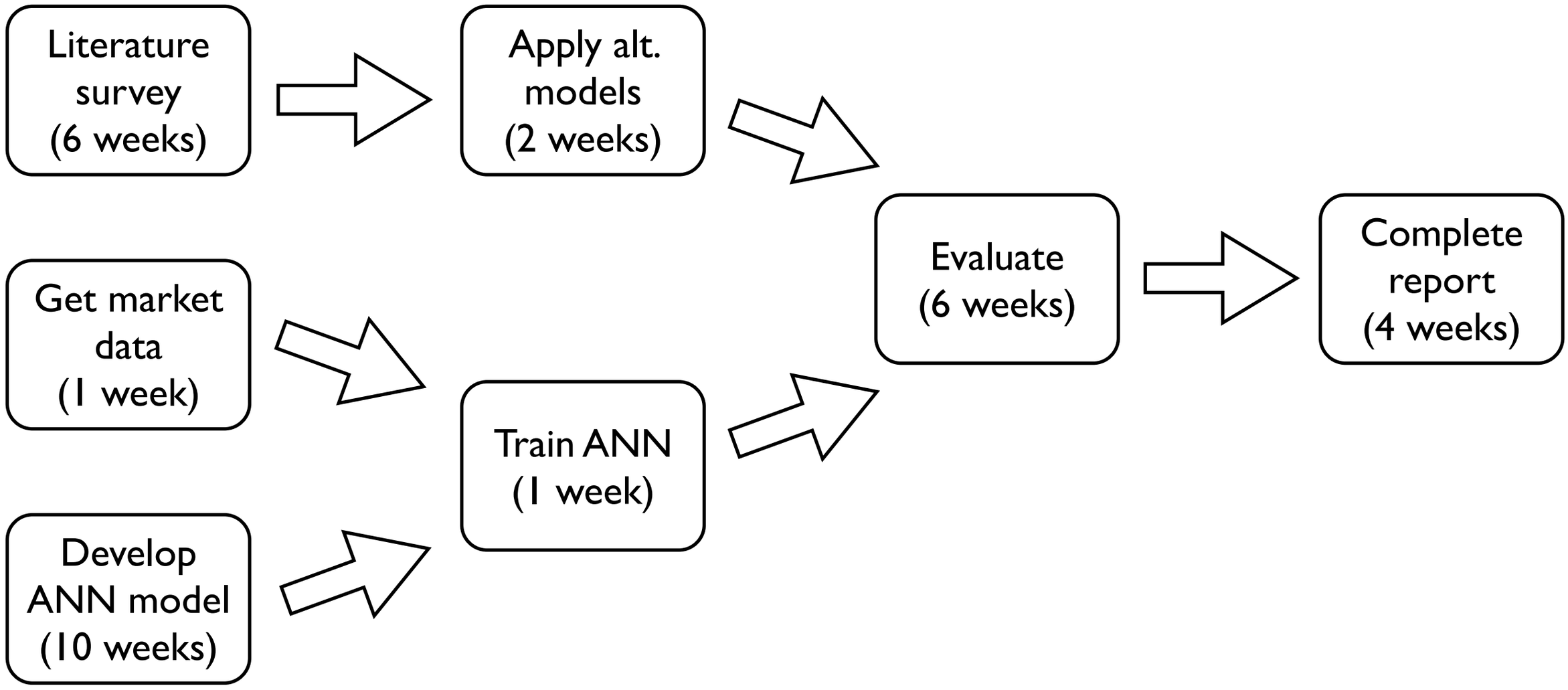
Time Estimates

Basic Activity	Estimate
Literature search	2 weeks
Literature review	4 weeks
Investigate ANN models	4 weeks
Design a suitable ANN	4 weeks
Implement and test the ANN model	2 weeks
Get stock market data	1 week
Train ANN on data	1 week
Apply alternative models to data	2 weeks
Identify statistical tests	2 weeks
Compare results using tests	4 weeks
Complete report	4 weeks
TOTAL	**30 weeks**

Activity Dependencies

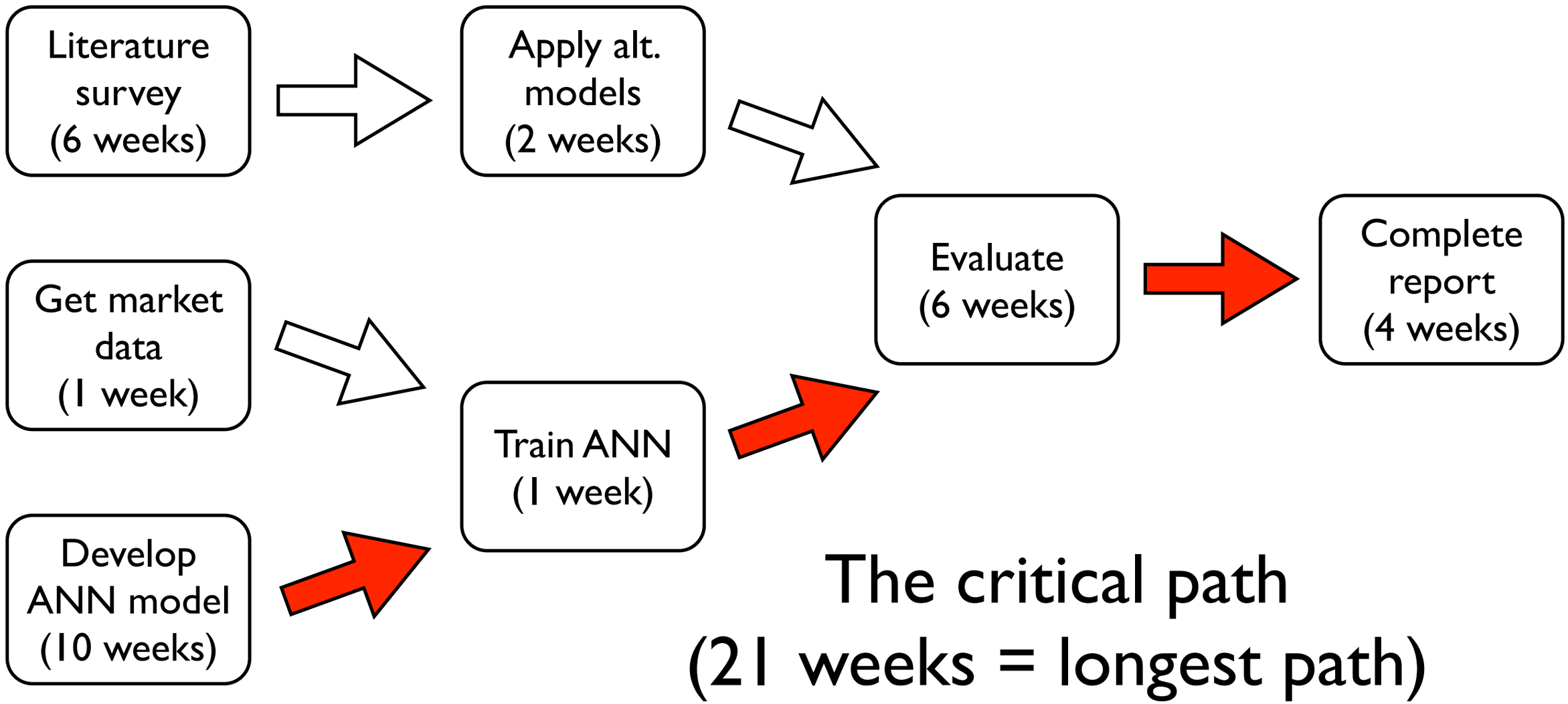
- **Logical** dependency: restriction on the order of activities.
    - E.g. activity A must happen before activity B.
- **Resource** dependency: an activity requires a limited resource.
    - E.g. we only have one programmer, so only one programming activity at a time.

Activity Diagram



Shows logical dependancies

Activity Diagram



The critical path
(21 weeks = longest path)

Critical Path Planning

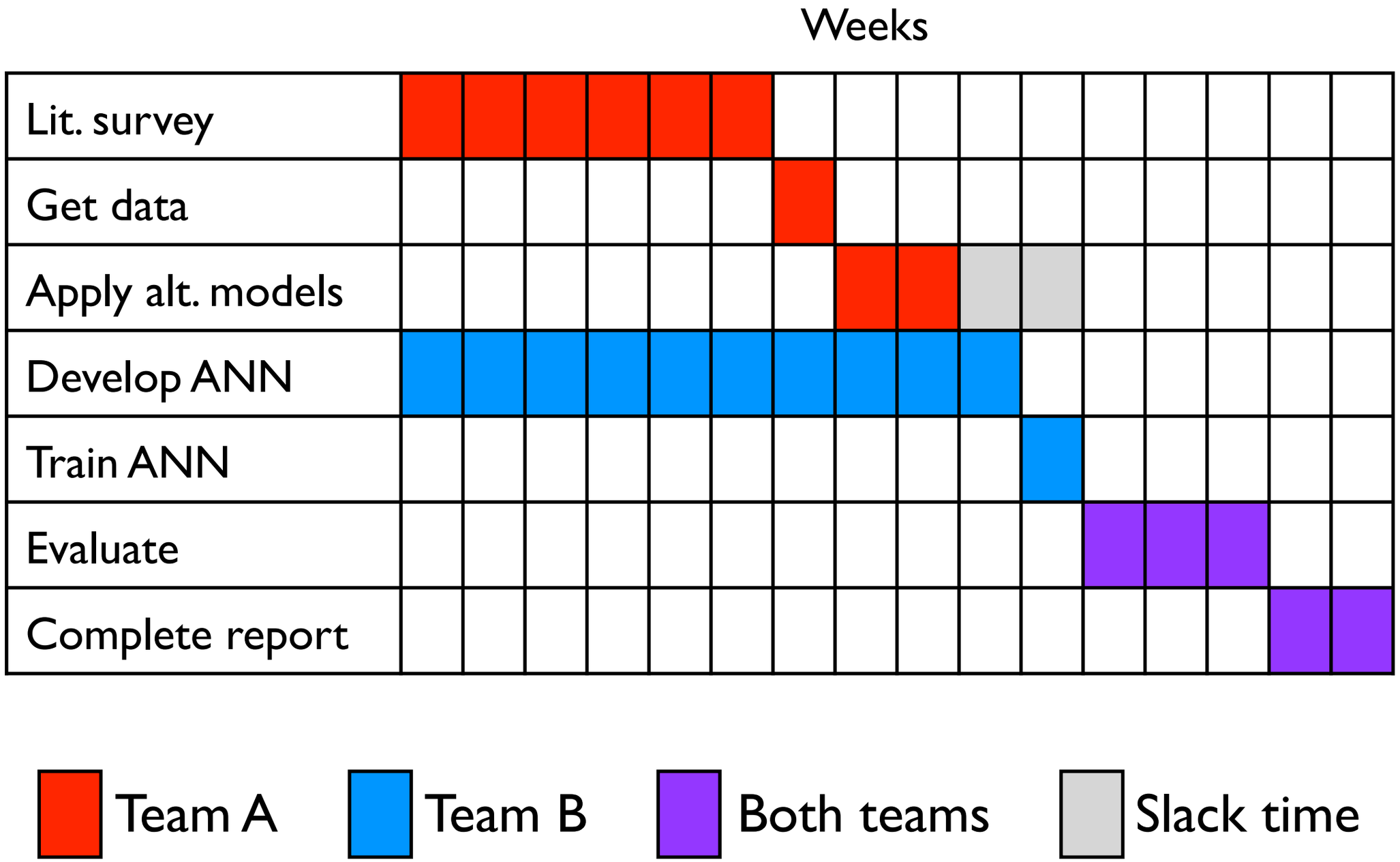
- The **critical path** is the sequence of activities which has the longest estimated time = the **minimum** project duration.
- Delays on the critical path will **hold up** the entire project. But increasing resources on the critical path can **speed up** the project.
- There can be more than one critical path, or near-critical paths, which deserve attention.
- Other (sub-critical) paths contain activities which could be delayed, if they're done in parallel. But delay them too long and they could become critical...

Scheduling

- The process of allocating **resources** to activities: time, people, computers, ...
- Typically represented in a **Gantt chart**.
- Must take account of dependencies:
  - Ensure correct activity order.
  - Ensure resources not over or under used, e.g. each team has enough work.

Gantt Chart

Stock Market Predictor



CEF Capital Schemes Gantt Chart - As at April 2008

Apr 08 BoD - Attachment 2

Feasibility | Briefing | Design | Tender | Construction | Commissioning

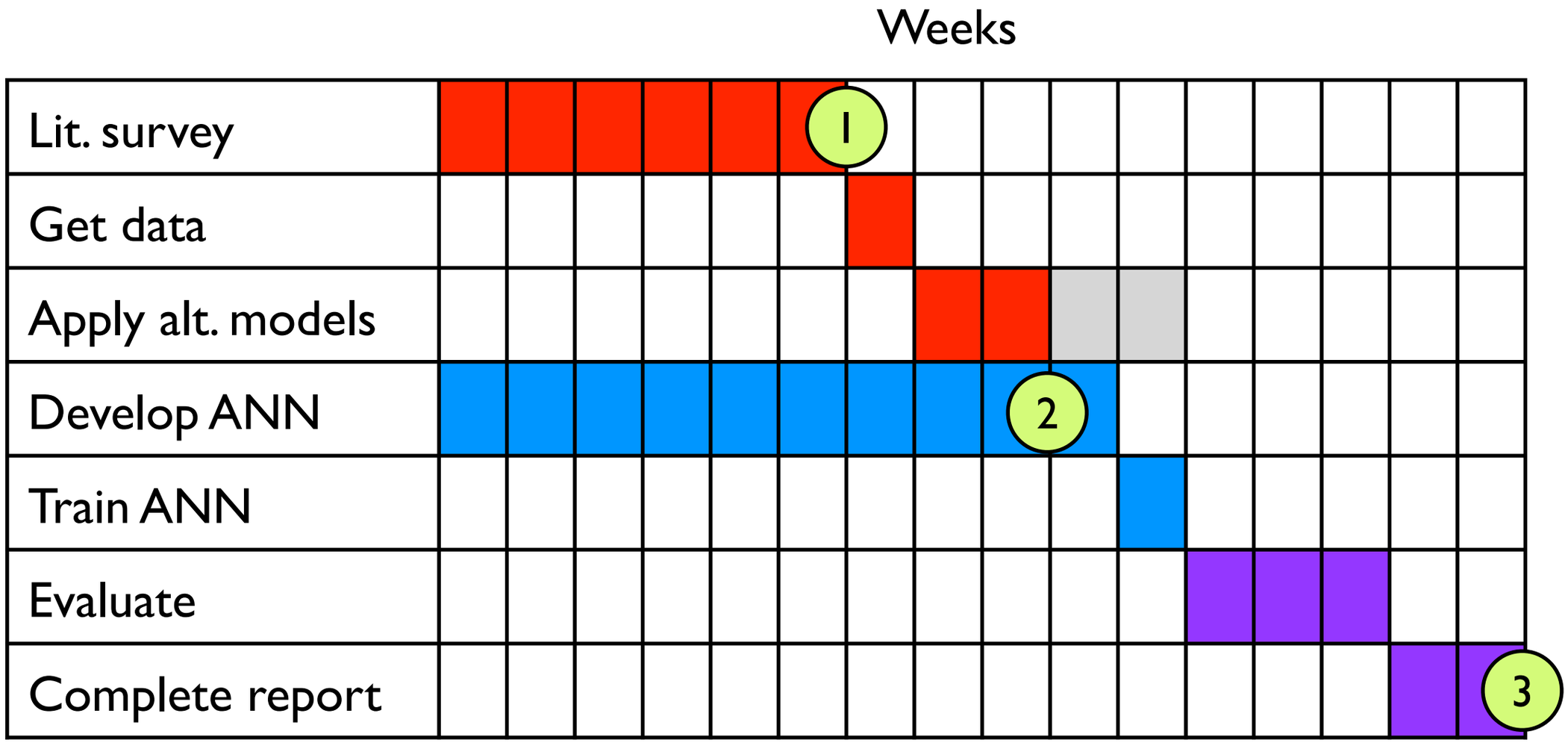
Line	Name	Jul-07	Aug-07	Sep-07	Oct-07	Nov-07	Dec-07	Jan-08	Feb-08	Mar-08	Apr-08	May-08	Jun-08	Jul-08	Aug-08	Sep-08	Oct-08	Nov-08	Dec-08	Jan-09	Feb-09	Mar-09
		2007						2008												2009		
	Estate Strategy (Designing the Future)																					
	Infrastructure Workstream																					
1	Guy's Tower Facade Feasibility										Draft										Design complete Aug 09	
2	East Wing Cladding (inc Ward Refurbishments)										Draft								Design stage to be complete by Oct 2009			
	Modernisation Workstream																					
3	Decan Ward @ STH (Feasibility & Design)																					
4	A&E Reconfiguration																					
5	St Thomas' Main Theatres Study																					
	Ambulatory Workstream																					
6	Mapping of Ambulatory Services																					
	Cancer Workstream																					
7	Redevelopment of Guy's Site incorporating Cancer																					
	Capital Development																					
8	Assisted Conception Unit																					
9	Bio Medical Research Centre & CRF (Guy's)																					
10	Bostock Ward																					
11	Cath Lab 3 - EW																					
12	Centralisation-Health Record Storage										Remaining programme to be confirmed											
13	CTU Cooling																					
14	ENT & Audiology Suite (Sth) - Phase 2										Programme to be confirmed											
15	Fluroscopy LW1										Remaining programme to be confirmed											
16	GLI Structural Monitoring & Repair										Programme to be confirmed											
17	HDR																					
18	Interim Arrangements for Samaritan Ward																					
19	Interventional Imaging at Guy's																					
20	IT Data Centre - Phase 2 Works																					
21	Kidney Treatment Centre - Tunbridge Wells																					
22	Labs: NW5 Enabling & CSR																					
23	Labs: Purine & Paediatric Labs - to NW4											(Phase 1)										
24	Major Incident Store (A&E)																					
25	New Laminar Flow Theatre at Guy's										Remaining programme to be confirmed											
26	North Wing Basement Entrance - Phase 1																					
27	North Wing Basement Entrance - Phase 2																					
28	Nuclear Medicine - Relocation of Scanner from St Thomas' to Guy's																					
29	Ophthalmology Back of Eye																					
30	Osteoporosis + Medical Photography Relocation (GT16 + GT13)							Scheme currently on hold														
31	Paediatric Neurosciences										Programme to be confirmed											
32	PET										Remaining programme to be confirmed											
33	Pneumatic Tube Upgrade																					
34	Pre-Operative Assessment Clinic at Guy's										Programme to be confirmed											
35	Pre-Operative Assessment Clinic at St Thomas'										Programme to be confirmed											
36	Radiology Environment & Waiting Areas (Imaging)																					
37	Relocate Palliative Care (GT13 to NGH F00)																					
38	Relocate Pharm Manf QC																			Expected to be complete by June 2009		
39	Relocation of Blundell Ward																					
40	Samaritan Ward - Bone Marrow Beds										Programme to be confirmed											
41	St Thomas HV Upgrade																					
42	St Thomas' Nursing Museum Feasibility Study																					
43	St Thomas' Retail Mall Feasibility Study																					
44	St Thomas' Theatre Upgrade Programme (9+10)																					
45	Surgical Admissions Lounge (inc Dimbleby)																					
	Engineering																					
46	AHU Serving Recovery - STH																					
47	BMS Updgrade to DESIEGO																					
48	Guy's Lifts 101-105 - Guys Tower																		Continued through to May 2010			
49	Guy's Lifts 110-114 - Guy's Tower																		Continued through to May 2010			
50	New Guy's House Lifts																					
51	NW Lifts																					
52	NW/LW Riser Refurbishment																		Continued through to June 2010			
53	Satchwell BMS Upgrade						On hold															
54	St Thomas' Chlorine Dioxide Water System							Remaining programme to be confirmed														

Milestones

- A **milestone** is a significant event in the project: an output or important moment for some activity (decision, report, software etc.)
- Project members can monitor progress by keeping an eye on milestones.
- A **deliverable** is a milestone output that is released outside the project.

Milestones

Stock Market Predictor

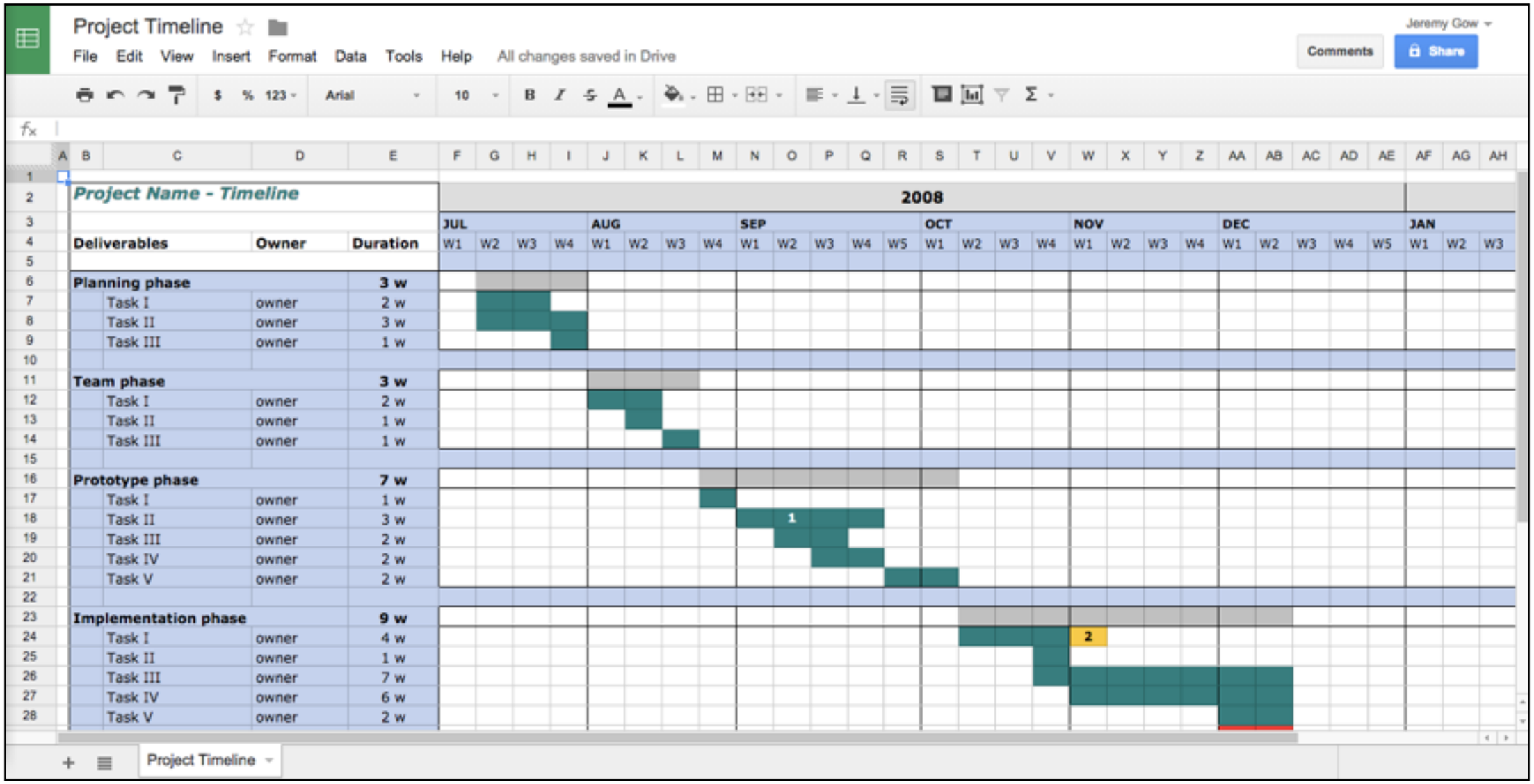


1 Literature report

2 Working ANN

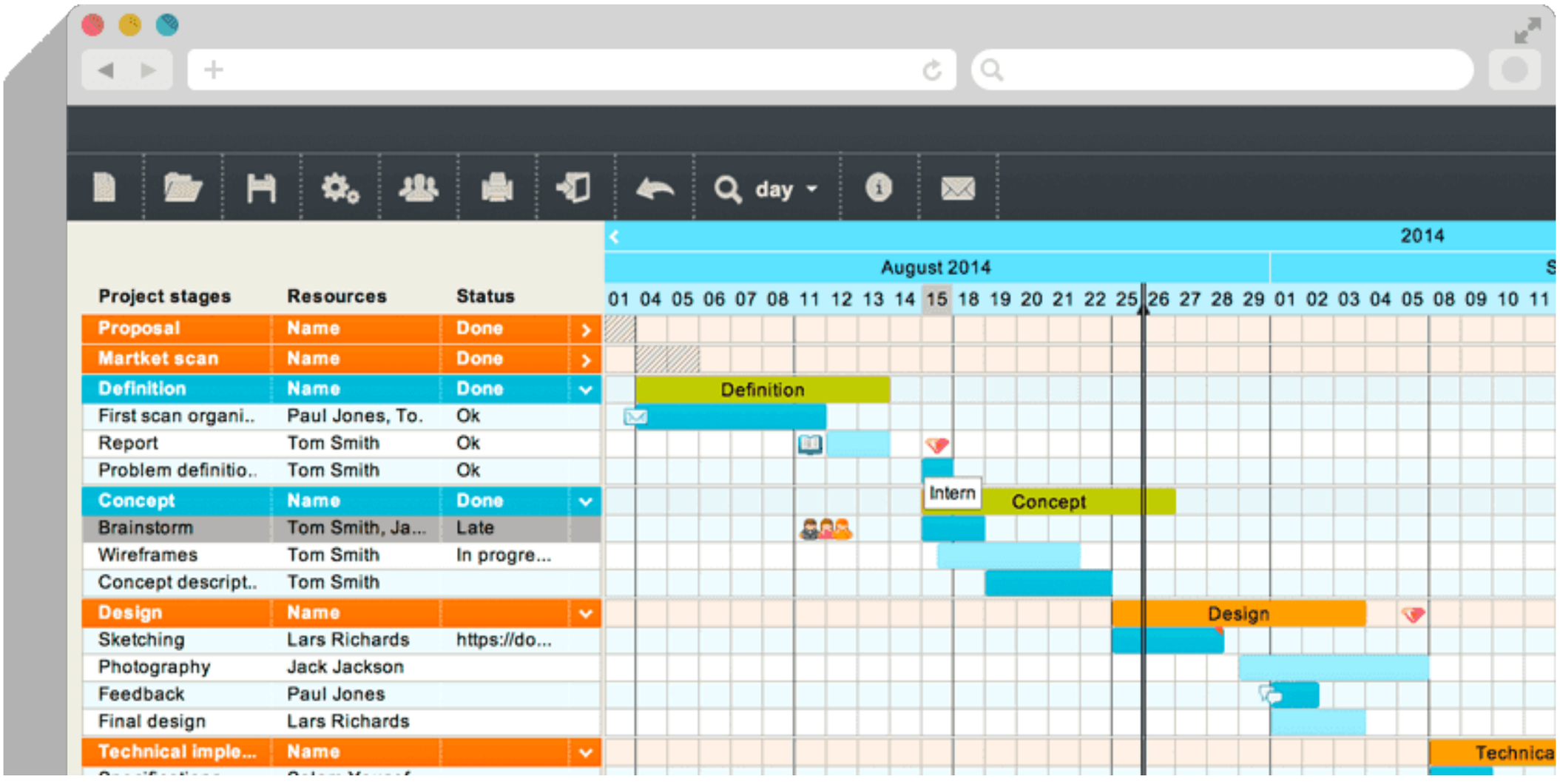
3 Final report

Free Online Resources



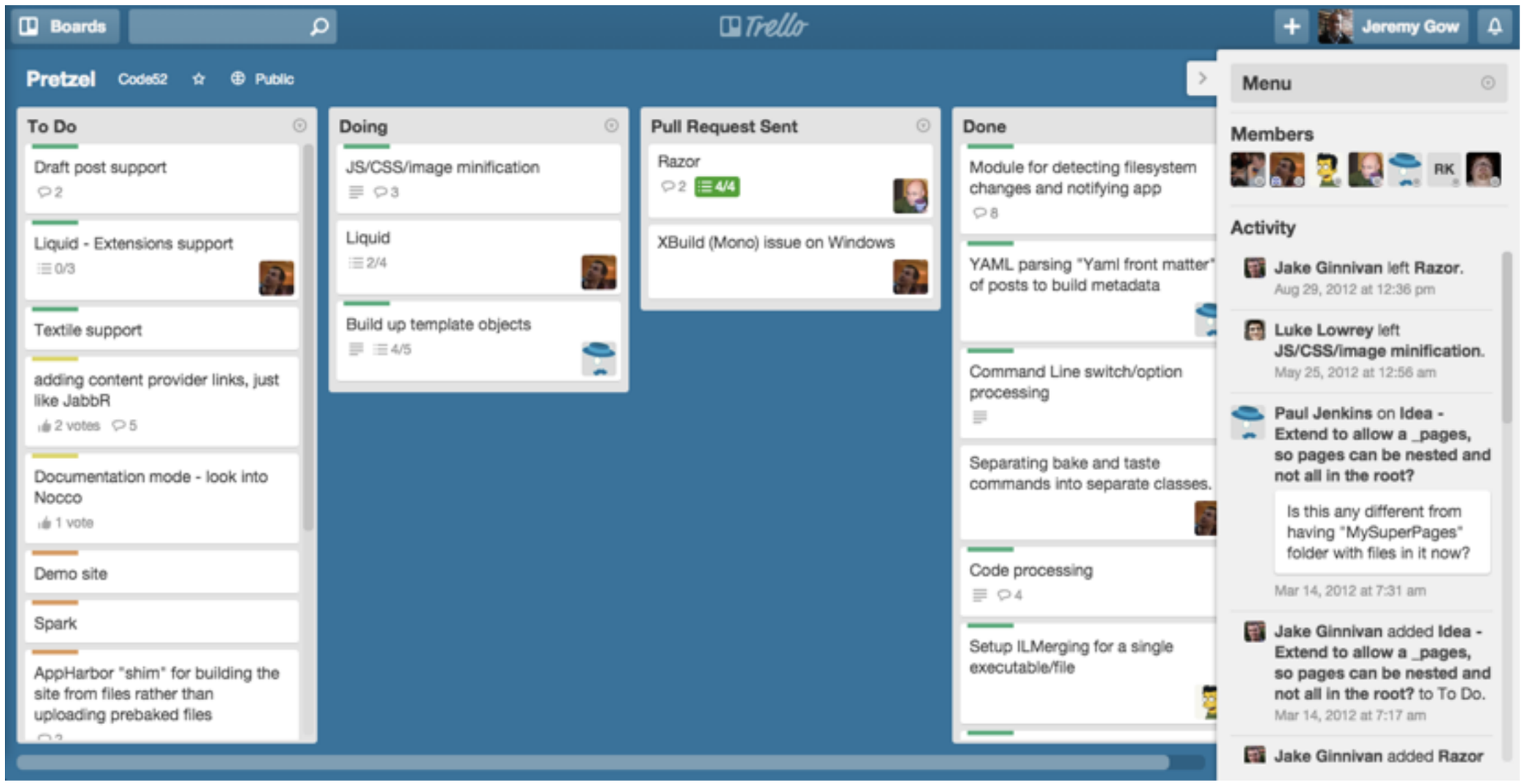
https://docs.google.com/

Free Online Resources



https://www.tomsplanner.com/

Free Online Resources



https://trello.com/

Next Steps

- Investigate free software for project planning.
- Start drafting a plan for your project: work breakdown, time estimates, activity diagram, Gantt chart and milestones.
- Separate plans for pre- and post-proposal phases.
- For managing larger commercial projects, see Pressman & Ince, chap. 7, "Project Scheduling and Tracking"

For Next Week

- Read "Scenarios" by Gerry Gaffney
  http://infodesign.com.au/wp-content/uploads/Scenarios.pdf
- Optional reading: Preece, Rogers & Sharp: chapter 10, "Identifying needs and establishing requirements"
- Meet with supervisors and discuss project ideas and requirements.