CMPT 373
Software Development Methods

Epilogue

Nick Sumner
wsumner@sfu.ca

What have we covered or refined?

- Building software
- Basic DevOps & testing
- Software processes (intent vs marketing)
- Software architecture

What have we covered or refined?

- Building software
- Basic DevOps & testing
- Software processes (intent vs marketing)
- Software architecture
- Designing to avoid complexity
- OOP
- Types, polymorphisms, & composition
- Design patterns (designing your own, not memorizing)

What have we covered or refined?

- Building software
- Basic DevOps & testing
- Software processes (intent vs marketing)
- Software architecture
- Designing to avoid complexity
- OOP
- Types, polymorphisms, & composition
- Design patterns (designing your own, not memorizing)
- Working with collections
- Designing better APIs
- Reasoning about correctness

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development
- In depth
  - Event driven design
  - Formal reasoning about software
  - Ambiguous & evolving requirements

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development
- In depth
  - Event driven design
  - Formal reasoning about software
  - Ambiguous & evolving requirements

- Polyglot & full stack dev.
- Domain driven design
- Data intensive applications

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development
- In depth
  - Event driven design
  - Formal reasoning about software
  - Ambiguous & evolving requirements

- Polyglot & full stack dev.
- Domain driven design
- Data intensive applications
- Auditability
- Team communication strategies
- Software estimation
- Scheduling & time management

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development
- In depth
  - Event driven design
  - Formal reasoning about software
  - Ambiguous & evolving requirements

- Polyglot & full stack dev.
- Domain driven design
- Data intensive applications
- Auditability
- Team communication strategies
- Software estimation
- Scheduling & time management
- Middleware systems

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development
- In depth
  - Event driven design
  - Formal reasoning about software
  - Ambiguous & evolving requirements

- Polyglot & full stack dev.
- Domain driven design
- Data intensive applications
- Auditability
- Team communication strategies
- Software estimation
- Scheduling & time management
- Middleware systems
- Career management

What have we *not* covered?

- Designing for
  - Reliability
  - Security
  - Performance
- Distributed systems
- Database systems
- Microservice architectures
- Web app development
- In depth
  - Event driven design
  - Formal reasoning about software
  - Ambiguous & evolving requirements

- Polyglot & full stack dev.
- Domain driven design
- Data intensive applications
- Auditability
- Team communication strategies
- Software estimation
- Scheduling & time management
- Middleware systems
- Career management

Attacking these in depth goes beyond what our prereqs allow

What you should look for

- Google site reliability engineering (SRE) resources [https://sre.google/books/]
  - Site Reliability Engineering
  - Building Secure & Reliable Systems
- Security Engineering [Anderson]

What you should look for

- Google site reliability engineering (SRE) resources **[https://sre.google/books/]**
  - Site Reliability Engineering
  - Building Secure & Reliable Systems
- Security Engineering **[Anderson]**
- Distributed application programming

What you should look for

- Google site reliability engineering (SRE) resources **[https://sre.google/books/]**
  - Site Reliability Engineering
  - Building Secure & Reliable Systems
- Security Engineering **[Anderson]**
- Distributed application programming
- Large scale software architecture

What you should look for

- Google site reliability engineering (SRE) resources **[https://sre.google/books/]**
  - Site Reliability Engineering
  - Building Secure & Reliable Systems
- Security Engineering **[Anderson]**
- Distributed application programming
- Large scale software architecture
- Web development

What you should look for

- Google site reliability engineering (SRE) resources **[https://sre.google/books/]**
  - Site Reliability Engineering
  - Building Secure & Reliable Systems
- Security Engineering **[Anderson]**
- Distributed application programming
- Large scale software architecture
- Web development
- Our courses
  - CMPT 431 (Distributed Systems)
  - CMPT 454 (Database Systems II)
  - CMPT 470? (Web dev?)
  - CMPT 474? (Cloud?)

What you should look for

- Google site reliability engineering (SRE) resources **[https://sre.google/books/]**
  - Site Reliability Engineering
  - Building Secure & Reliable Systems
- Security Engineering **[Anderson]**
- Distributed application programming
- Large scale software architecture
- Web development
- Our courses
  - CMPT 431 (Distributed Systems)
  - CMPT 454 (Database Systems II)
  - CMPT 470? (Web dev?)
  - CMPT 474? (Cloud?)

If you hear a student say
"Don't take YYY, it's hard."
that is probably good to take.

Planning for the future

- Be aware of
  - the job market & competition
  - your circumstances (e.g. Covid + Market)
  - how hiring works

Planning for the future

- Be aware of
  - the job market & competition
  - your circumstances (e.g. Covid + Market)
  - how hiring works
- Practice & Develop side projects

Planning for the future

- Be aware of
  - the job market & competition
  - your circumstances (e.g. Covid + Market)
  - how hiring works
- Practice & Develop side projects
- Specialize (our industry is an iceberg)

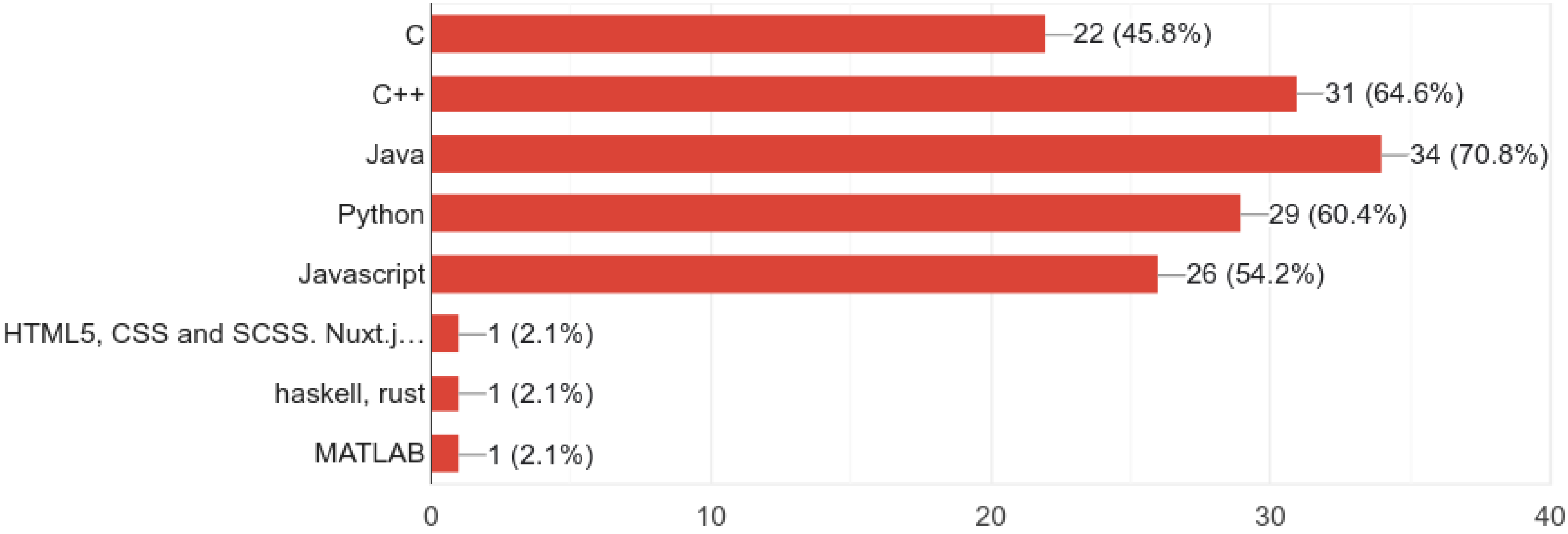
Planning for the future

- Be aware of
  - the job market & competition
  - your circumstances (e.g. Covid + Market)
  - how hiring works
- Practice & Develop side projects
- Specialize (our industry is an iceberg)
- Identify your weaknesses and create a plan to improve them

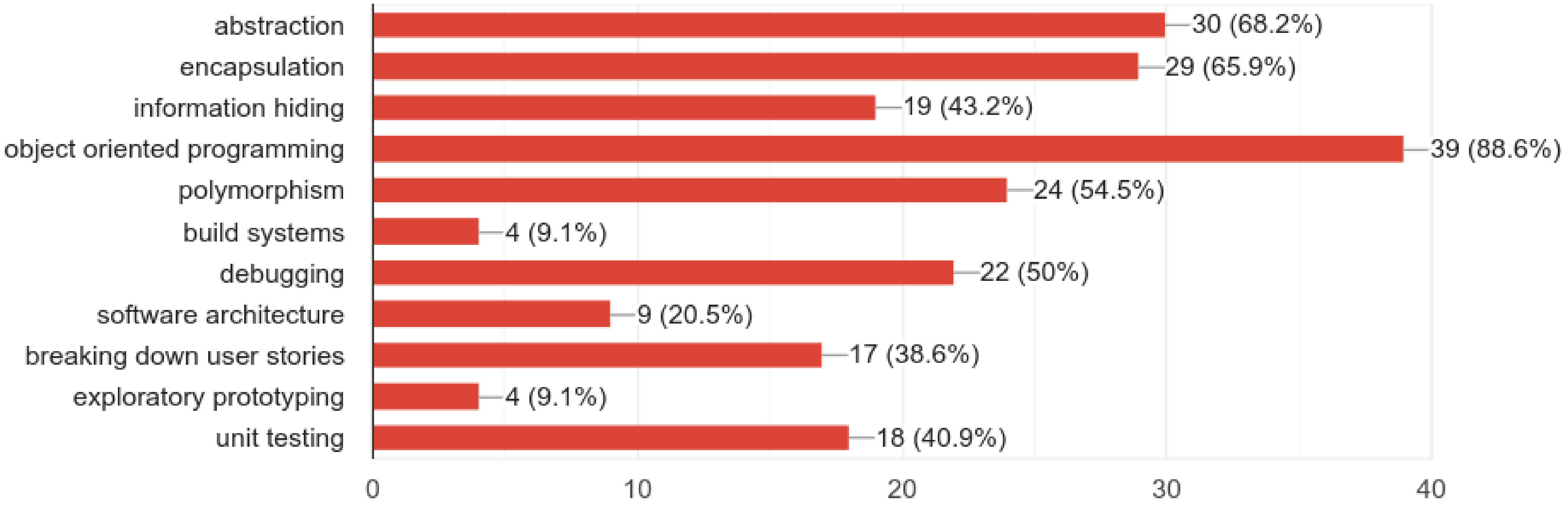
Planning for the future

- Be aware of
  - the job market & competition
  - your circumstances (e.g. Covid + Market)
  - how hiring works
- Practice & Develop side projects
- Specialize (our industry is an iceberg)
- Identify your weaknesses and create a plan to improve them
- Train your metacognition

Planning for the future



Planning for the future



Onward

- While many struggled, I have seen people adjust & improve.

Onward

- While many struggled, I have seen people adjust & improve.
- Keep in mind that however daunting the circumstances,
  you can affect the outcome.

Q & A