

Beginning to Program

Computer & Information Sciences

W. H. Bell

Overview

- Motivation.
- Capturing requirements.
- Forming a design.
- Ordering thoughts.
- Writing software.
- Summary.

Motivation

- Software is driven by requirements.
- Problems that cannot be solved efficiently with existing tools.
- A new set of ideas or a new process.
- A new computer or supporting set of software packages.
- Software can allow ideas to become realities.
- Release the potential of users.
- Deliver savings and a better service.

First thoughts

Solve my problem!



Capturing user requirements

User interface



Data flow



Data flow

User interface



14:20:10 --- Working

Capturing user requirements

User interface



Data flow

- Read data
- Handle user inputs
- Process values
- Provide user outputs
- Write data



Data flow

User interface

14:20:10 --- Working

Minimum viable product

User interface



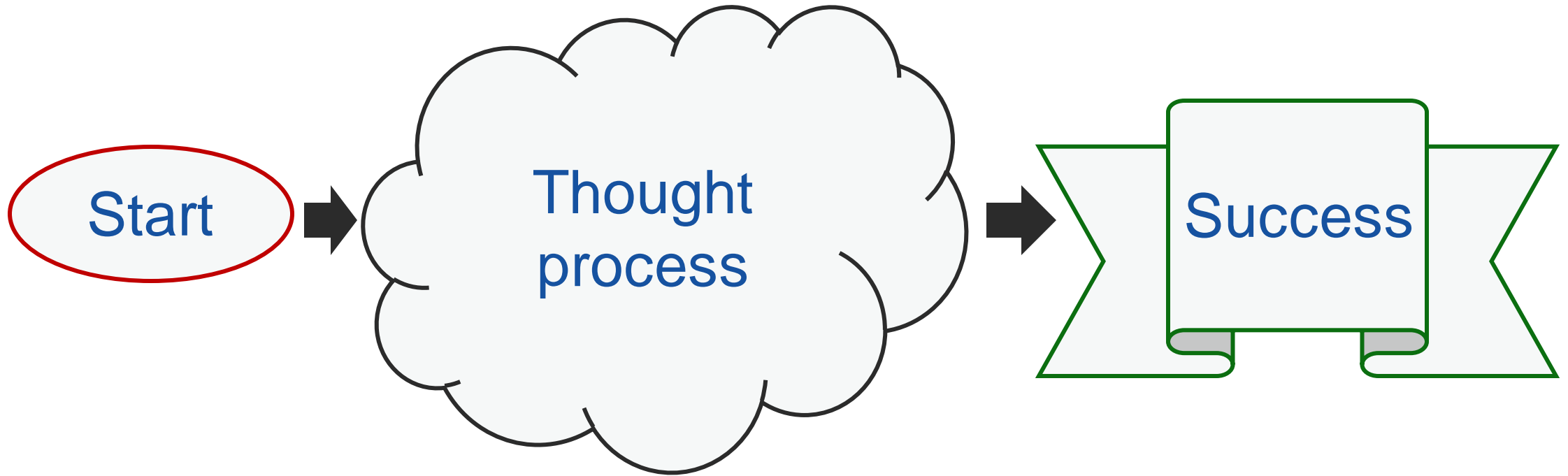
Data flow

- Read data
- Handle user inputs
- Process values
- Provide user outputs

User interface

14:20:10 --- Working

Reaching the goal



Need to break thought process down into steps.

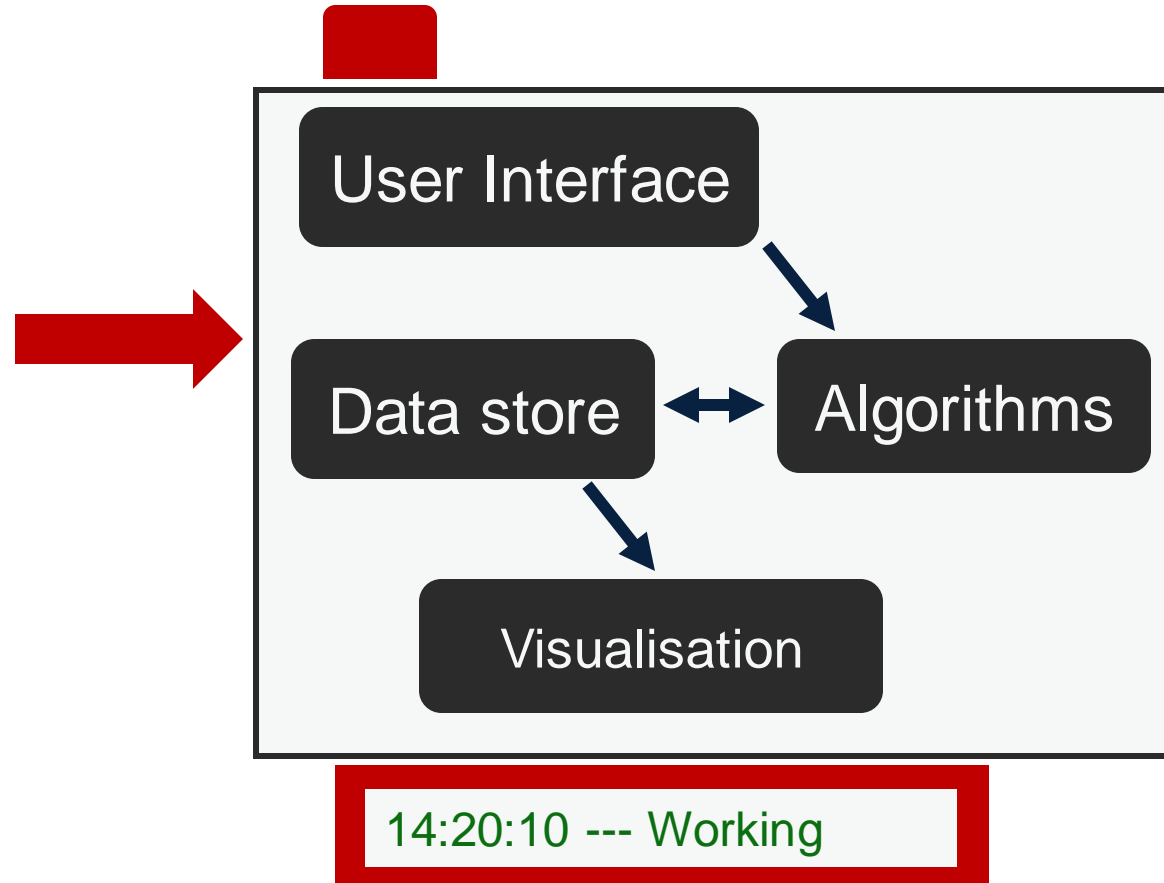
Need a design that contains these steps.

Need interfaces between design components.

Architecture

- What type of solution is needed?
- Simple algorithm.
- Desktop program.
- Client and server.
- Web service.
- Running in web browser.
- Multithreaded or single process.
- Data storage.
- Security requirements.

High-level design



Low-level design

- Interfaces between components.
- Algorithm specification.
- Data description.
- Example uses and processes.

Ordering thoughts

- Flowcharts and process flow.
- Can be useful for algorithms or for conveying high-level requirements.
- Pseudocode.
- Create structure in language that is not final code.
- Fast and can re-use for source code comments.
- Record thoughts.
- Break program down into small steps that be implemented easily.
- Re-use for documentation.

Write in units

- A function should have a clear purpose.
- Short enough to understand and debug.
- Data objects (classes) for a particular role.
- Collection of data values.
- Functions to access them.
- Clear data flow, from one unit to another.
- Avoid copying data around or modifying them in many places.

Write and test

- Write software and then test it.
- Developer tests.
- Unit tests.
- Testing is needed to provide confidence in software.
- Many users involved in development.
- Development over a long period of time.

Language specific

- Implementation is programming language specific.
- Understand specific syntax and solutions for a language.
- Learn from other developers.
- May need to experiment.
- Build test code to try out ideas.

Summary

- Provided motivations for computer programming.
- Discussed the process of software development.
- Presented some mechanisms for ordering thoughts.
- Suggested how programs should be constructed.