**Project Design & Implementation**

- **Where do I begin?**
  - Modularity
    - Top Down Design
    - Testing
    - Timetabling
    - Partial Completion

**Testing**

- Test each module individually
  - If things work individually, they’ll probably work together
  - If not, it’s easier to fix small modules
  - Write the tests as you go along
- Keep testing
  - Test Harness
  - Extend the tests as you uncover problems
- Don’t leave testing till the end
- Test in the final environment
  - Don’t assume it’ll be OK if it works on your laptop

**Top Down Design**

- Make it simpler
  - Turn a big problem into lots of little ones
  - If you’re not sure what to do now, decide what to do next
  - “Never do today what you can put off till tomorrow”
- Not Just for Writing Programs
- Dave’s Plan for a Perfect World
  - Take over the world
  - Establish Universal Peace
  - Retire
  - Learn to Ski

**Timetabling**

- Modules help in drawing up a timetable

<table>
<thead>
<tr>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>World Domination</td>
<td>Universal Peace</td>
<td>Run</td>
<td>Learn to Ski</td>
</tr>
</tbody>
</table>

- Adjust the timetable if necessary

<table>
<thead>
<tr>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>World Domination</td>
<td>Universal Peace</td>
<td>Run</td>
<td>Learn to Ski</td>
</tr>
</tbody>
</table>
Partial Completion

- If you run out of time
- Prioritise and complete selected modules

<table>
<thead>
<tr>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>World Domination</td>
<td>Universal Peace</td>
<td>Apollo</td>
<td>Learn to Ski</td>
</tr>
</tbody>
</table>

- Half a program that works – OK to demonstrate
- A program that half-works – much less useful

Functional Modularity

- e.g. Web-fronted Database
  - Web interface
  - Database backend
  - Customer view
  - Administration view
- Develop separately then combine

Prototype the form and output – call for feedback
- e.g. Use fixed results to illustrate the interface
- Don’t need to query the database initially

Design Adaptation

- Feedback from prototypes ⇒ changes to original design
- Modular design limits the pain
Regular Snapshots

- If things need to be changed
  - When things need to be changed
  - Modular design limits the pain
- “Write it twice”
  - Second revision is invariably better
  - Rewriting a module is realistic
  - Rewriting a whole program is not

Don’t decide this yet
- Don’t even think about it yet!
- Minimal impact on design
- Design should drive development choices
  (not the other way around)
- Modularity allows easy change
- A clean design is adaptable
  - Web interface / Standalone App
  - Same underlying processing
  - Just a different interface

Choice of Development Environment

What you already know
- What you would like to know
- What is available to you
- What is available to the client
- What the client can support
- What is suitable for the task

Development Environment Considerations

Write it down
- Write it down now
- Write everything down
- Write down why

Documentation

- Revision control
  - “It used to work, now something’s changed & it’s broken”
  - Regular Snapshots
  - Full revision control - snapshots of all changes
  - Continuous documentation

Maintenance
<table>
<thead>
<tr>
<th>Communication</th>
<th>Summary</th>
</tr>
</thead>
</table>
| ✦ Talk to your client  
✦ Talk to us  
✦ Talk to each other | ✦ Modules, Modules, Modules!  
✦ Work on stuff in parallel  
✦ Don't let one problem stop everything  
✦ Don't leave everything to the end  
✦ Particularly testing and documentation |

<table>
<thead>
<tr>
<th>Detailed Topics for Another Time</th>
</tr>
</thead>
</table>
| ✦ Department/University Services  
  ✦ WebMail  
  ✦ Remote Access  
  ✦ Web & Database facilities  
  ✦ Personal systems  
  ✦ Web & Database facilities  
  ✦ WAMP/LAMP/XAMP  
  ✦ Revision Control  
  ✦ Programming  
  ✦ Debugging techniques  
  ✦ Portability  
  ✦ What else do you want to know? |