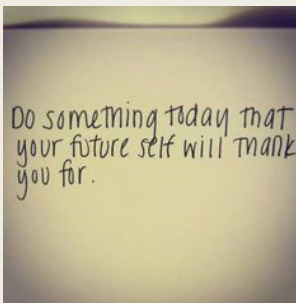


PROGRAMMING: BEST PRACTICES

Part I - The Big Picture





[2] Backward Planning

- Used to make sure your projects are completed on time

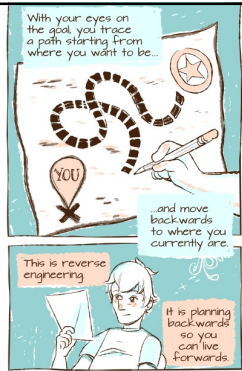


[2] Backward Planning

a.k.a

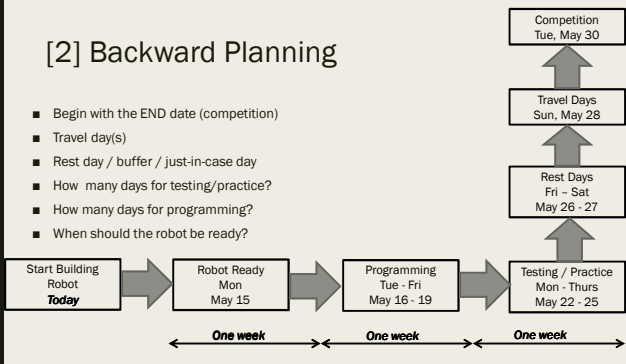
Reverse Engineering

Planning backwards so you can live your life forwards



[2] Backward Planning

- Begin with the END date (competition)
- Travel day(s)
- Rest day / buffer / just-in-case day
- How many days for testing/practice?
- How many days for programming?
- When should the robot be ready?




REVIEW...



The first poster features a blue silhouette of a person with a smaller person inside, holding a laptop. The text reads "BE KIND TO YOUR FUTURE self". The second poster shows a person reading a document with a path leading to a goal. The text reads "With your eyes on the goal you trace a path starting from where you want to be." and "Planning backwards so you can live your life forwards".

[3] Two Programmers

- At least 2 on the team who can program
- One may be the expert (main programmer) but the other person should understand the code reasonably well enough to modify it if needed.



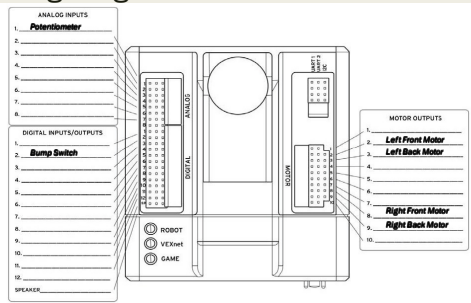
The illustration shows two people sitting at desks with computers. Above them are various icons representing different aspects of technology and programming, such as a cloud, a music note, a gear, and a lightbulb.

REVIEW...



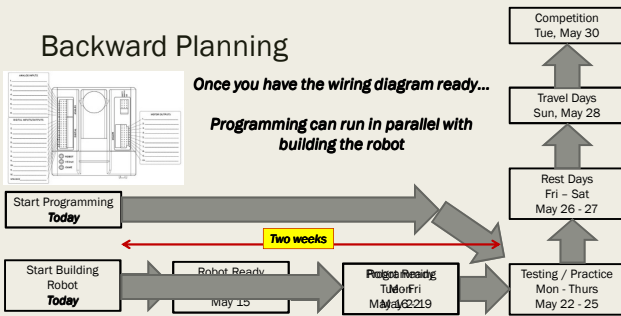
The first poster is the same as in the first slide. The second poster is titled "Backward Planning" and features the same text and graphics as the second poster in the first slide. The third poster is the same as the illustration in the second slide.

[4] Wiring Diagram

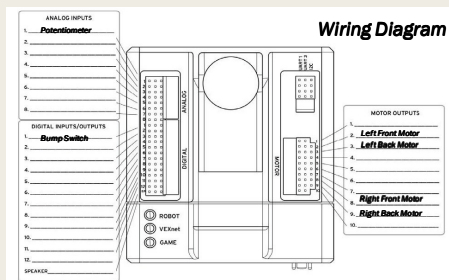


Backward Planning

Once you have the wiring diagram ready...
 Programming can run in parallel with building the robot



REVIEW...



BE KIND TO YOUR FUTURE SELF

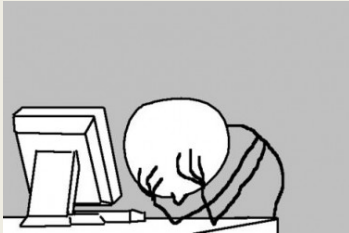
Backward Planning

Planning backwards so you can live your life forwards

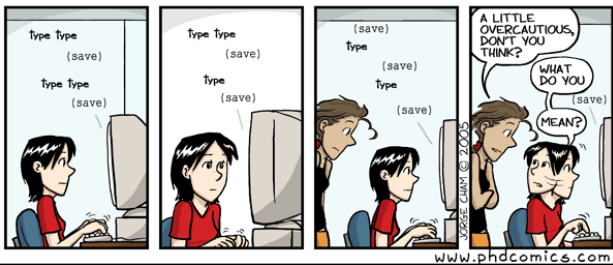
Two Programmers

23F

[5] Backup your code – Don't let this happen to you (no backup)



[5] Backup your code – save often



[5a] Backup your code locally
(i) computer,
(ii) USB drive

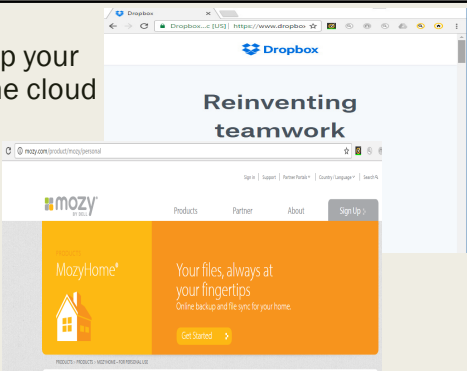


- *VexWorlds_1_17_2017.c*
- *VexWorlds_1_23_2017.c*
- *VexWorlds_2_09_2017.c*

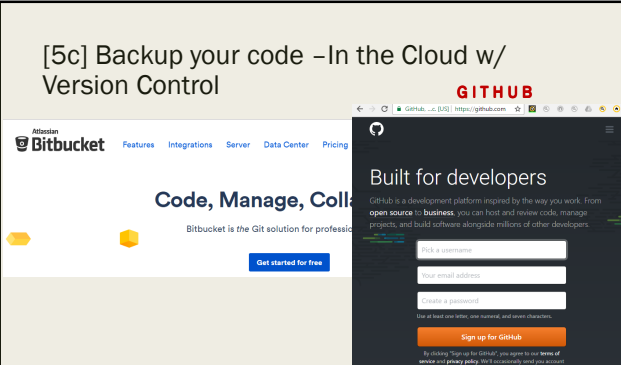


[5b] Backup your code - In the cloud

- **Dropbox**
- **Mozy**



[5c] Backup your code - In the Cloud w/ Version Control



The image shows two overlapping website screenshots. On the left is the Bitbucket website with the headline 'Code, Manage, Collaborate' and a 'Get started for free' button. On the right is the GitHub website with the headline 'Built for developers' and a sign-up form with fields for 'Pick a username', 'Your email address', and 'Create a password', followed by a 'Sign up for GitHub' button.

[5c] Why Version Control (VC)?

1. Clutter free workspace (no numbered files)
2. Backup and restore
3. Collaboration
4. Track changes (with messages)
5. Track ownership

