

# Software Development – Basics

---

- ▶ Not talking about *programs*, but about *program system products*
- ▶ Know about programs, what about PSPs?

# Software Development – Basics

- ▶ Program: complete by itself
- ▶ Ready to run:
  - By author only
  - For planned inputs only
  - On system on which it was developed



# Program System

---

- ▶ Each program is a component in integrated collection
- ▶ Precisely defined interface: all programs comply
- ▶ Each program sticks to reasonable resources
- ▶ Each program is tested with other programs, many combinations

# Program Product

---

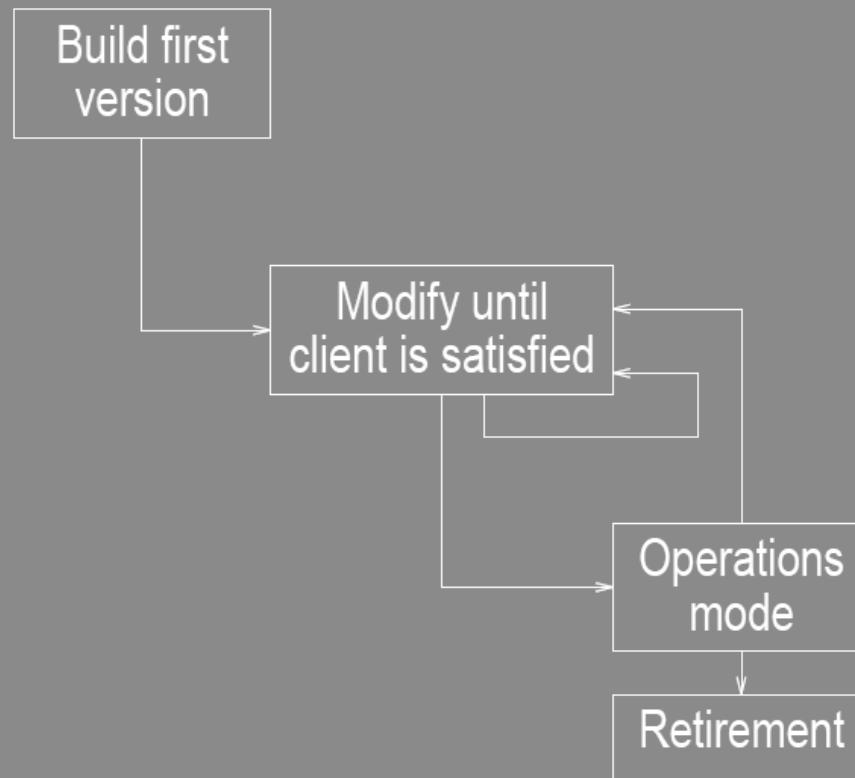
- ▶ Product can be run, tested, repaired, extended by anyone
- ▶ Runs on multiple platforms
- ▶ Accommodates many sets of data
- ▶ Generic range and form of input
- ▶ Provides response to invalid inputs
- ▶ Provides documentation for users and developers

# Program Product

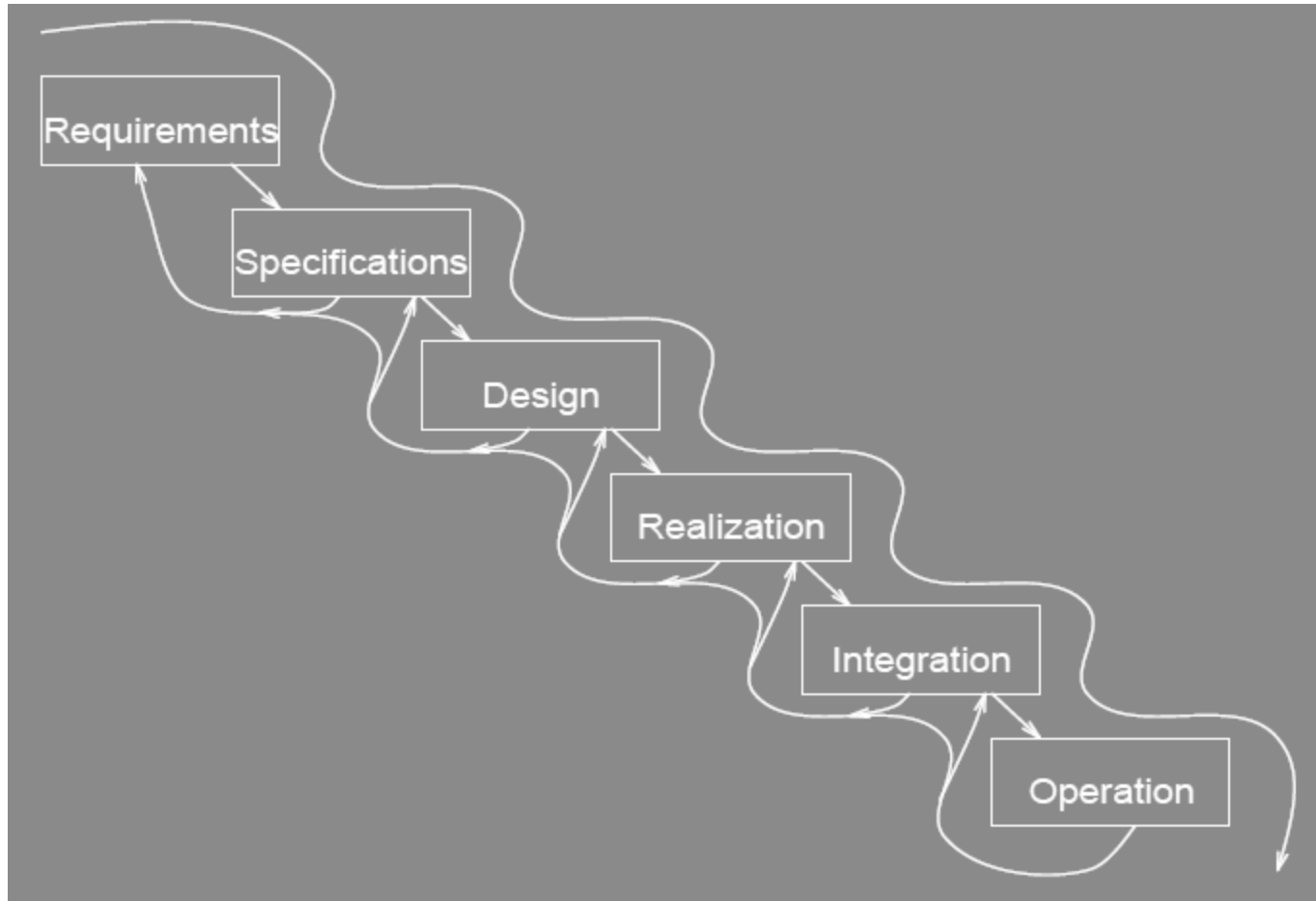
- ▶ We think about programs, instead of PSPs.
- ▶ Affects all calculations:
  - Ease vs. difficulties
  - Time
  - Costs
- ▶ All off by a magnitude

# Common SPM models: Build-N-Fix

## Build-and-Fix

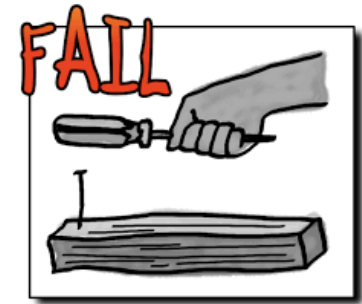


# Common SPM models: Waterfall cycle



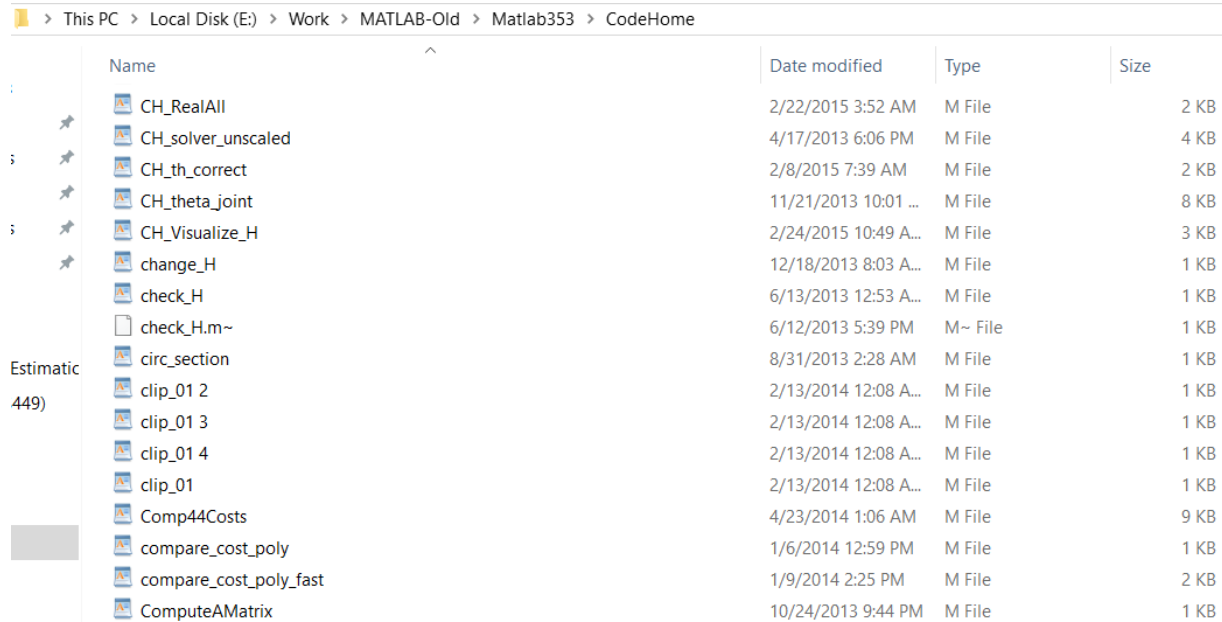
# A software developers toolkit: Which ones do you have?

- ▶ **IDE and debugger**
- ▶ **Compare files and find differences**
- ▶ **Work with regular expressions**
- ▶ **Source control**
- ▶ **Versioning and sharing**
- ▶ **Software testing**





# I know you too have this on your drive



The screenshot shows a Windows File Explorer window with the address bar path: This PC > Local Disk (E:) > Work > MATLAB-Old > Matlab353 > CodeHome. The main pane displays a list of files with columns for Name, Date modified, Type, and Size. The files are listed in descending order of date modified. The left sidebar shows the navigation pane with 'Estimate' and '449)' visible.

Name	Date modified	Type	Size
CH_RealAll	2/22/2015 3:52 AM	M File	2 KB
CH_solver_unscaled	4/17/2013 6:06 PM	M File	4 KB
CH_th_correct	2/8/2015 7:39 AM	M File	2 KB
CH_theta_joint	11/21/2013 10:01 ...	M File	8 KB
CH_Visualize_H	2/24/2015 10:49 A...	M File	3 KB
change_H	12/18/2013 8:03 A...	M File	1 KB
check_H	6/13/2013 12:53 A...	M File	1 KB
check_H.m~	6/12/2013 5:39 PM	M~ File	1 KB
circ_section	8/31/2013 2:28 AM	M File	1 KB
clip_01 2	2/13/2014 12:08 A...	M File	1 KB
clip_01 3	2/13/2014 12:08 A...	M File	1 KB
clip_01 4	2/13/2014 12:08 A...	M File	1 KB
clip_01	2/13/2014 12:08 A...	M File	1 KB
Comp44Costs	4/23/2014 1:06 AM	M File	9 KB
compare_cost_poly	1/6/2014 12:59 PM	M File	1 KB
compare_cost_poly_fast	1/9/2014 2:25 PM	M File	2 KB
ComputeAMatrix	10/24/2013 9:44 PM	M File	1 KB

## A dump of code

What to do with it? **Close this window and NEVER come back**

# A slightly better approach for beginners

- ▶ “I know I should comment, write good code etc. *but I didnt.*”
- ▶ **How to salvage something out of this dumpster?**

## Here is your way backwards

- ▶ When your project ran, always save that version as \_EXE(even when its due tomorrow)




- ▶ After you are done with submission etc., go to \_EXE file
- ▶ Use a dependency tracker to find all related code, then copy it to a new folder
- ▶ Add documentation, and check out on Git.

# Looks better

» This PC » Local Disk (E:) » Work » MATLAB-Org » Exchange » MBIR » PlugAndPlayCH\_v1 »

Name	Date modified	Type	Size
ComputeSurrogateError	1/1/2017 12:41 AM	M File	1 KB
ComputeAMatrix	1/1/2017 12:41 AM	M File	1 KB
Cost_DataTermAL	1/1/2017 12:41 AM	M File	1 KB
DataTermOpt	1/13/2017 2:50 PM	M File	3 KB
Denoise_AL_Poly_For_PnP_Java128_v3_HU...	1/1/2017 12:41 AM	M File	4 KB
denoise_cost_Hf_impl_1_0528_01	1/1/2017 12:41 AM	M File	2 KB
FitInVessel	1/1/2017 12:41 AM	M File	1 KB
format_image_for_publication	1/1/2017 12:41 AM	M File	1 KB
forward_project_v2	1/1/2017 12:41 AM	M File	1 KB
GenDenoisingCostsPlot	1/1/2017 12:41 AM	M File	3 KB
GetTitle	1/1/2017 3:09 PM	M File	1 KB
GetUniformAngles	1/1/2017 12:41 AM	M File	1 KB
HomogenousUpdate	1/1/2017 12:41 AM	M File	4 KB
HPlot	1/1/2017 3:09 PM	M File	1 KB
IDisp	1/1/2017 12:41 AM	M File	1 KB
ImageInterp3D	1/1/2017 12:41 AM	M File	1 KB
InterleaveRecon	1/1/2017 12:41 AM	M File	1 KB
InterleaveReconCustom	1/1/2017 12:41 AM	M File	1 KB
JDenoise_CH.asv	6/25/2017 3:21 AM	ASV File	3 KB
JDenoise_CH	1/13/2017 4:01 PM	M File	2 KB
LoadAlloyData	1/13/2017 6:45 PM	M File	1 KB
MakeVideo	1/1/2017 3:09 PM	M File	4 KB
perform_tv_denoising	1/1/2017 3:09 PM	M File	6 KB
PlotPnPCost	1/1/2017 3:09 PM	M File	3 KB
PlugAndPlayInitCustomGeomSubsFewerIt...	1/1/2017 3:09 PM	M File	3 KB

# Much better

 / **FSGroupSurfTtest**

Watch 0 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Settings](#) [Insights](#)


This MATLAB script performs comparison between GM surface thickness of two populations of subjects that have already been processed using FreeSurfer anatomical processing pipeline(recon-all) [Edit](#)

[Add topics](#)

4 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

	Update README.md	Latest commit 14bf331 on Jun 26
<a href="#">BatchSurfTest.m</a>	Just corrected some structure-same code	3 months ago
<a href="#">FreeSurfGroupTtest.sh</a>	Just corrected some structure-same code	3 months ago
<a href="#">README.md</a>	Update README.md	3 months ago
<a href="#">SurfTtest.m</a>	Just corrected some structure-same code	3 months ago

 README.md

## FSGroupSurfTtest

This MATLAB script performs comparison between GM surface thicknesses of two populations of subjects that have already been processed using FreeSurfer anatomical processing pipeline(recon-all)

You need MATLAB and FreeSurfer to execute this code.

Sample usage(MATLAB script): `SurfTtest(P1,P2,'P1vsP2','P1','P2','PN');`

P1 and P2 are the names of the subjects in two groups being compared. These have to be MATLAB strings. Each of these subjects must have already been processed using FreeSurfer processing pipeline(recon-all or similar) and their surface stats must be present in \$FREESURFER\_DIR/subjects/Sub1/surf/ directory.

The directory 'P1vsP2' contains all the intermediate/FSGD etc files created.

# Using source control–steps

Connect local code  
directory with Git repo  
**Commit** code



Create **Github** account  
Create new rep on Github  
Install **TortoiseSVN**

Checkout from Git  
to **make working  
copy**  
Work on it  
**Commit** and add  
comments

*“Code complete, A practical handbook of software construction”* Steve McConnell

US



# Some tips to write professional code

- ▶ 1: Don't write code
  - I/O, sort, search, webIO, images...
  - Instead, reuse...



*"Code complete, A practical handbook of software construction"* Steve McConnell

# Some tips to write professional code

- ▶ 2: Don't do hard things, do easy things
  - Avoid tricky algorithms that you don't understand
  - Create a new module for a new task
- ▶ 3: Re-factoring > Rewriting
- ▶ 4: Rewriting > Patching

*"Code complete, A practical handbook of software construction"* Steve McConnell

