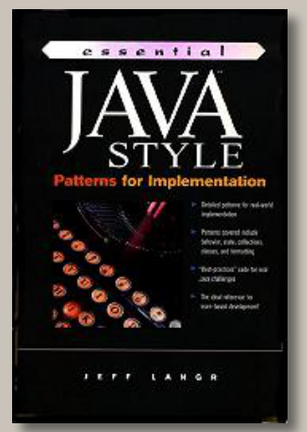
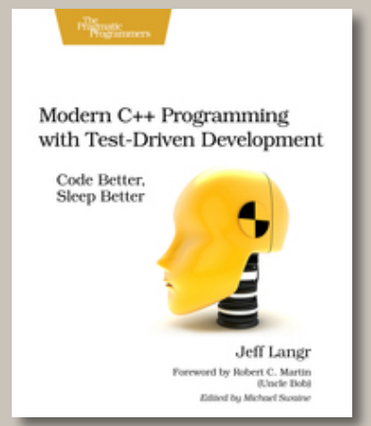
From Mob to Symphony

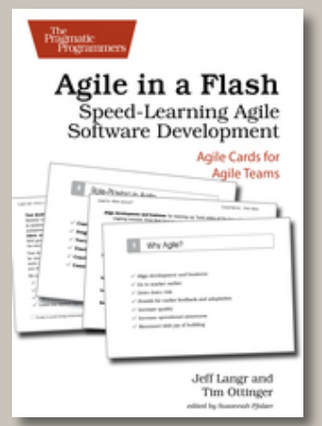
Langr
Software Solutions
SOFTWARE TRAINING/CONSULTING
TDD
AGILE

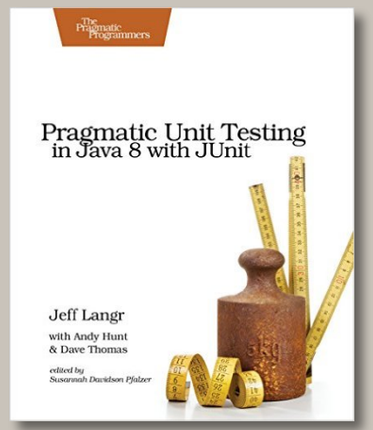
A Mob Composing Workshop

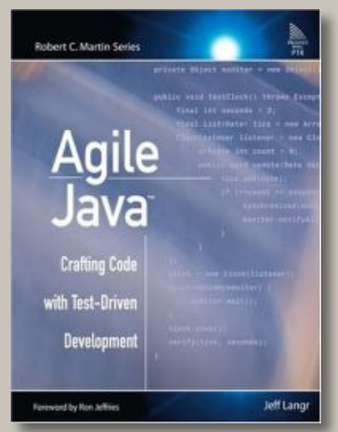
jeff@langrsoft.com
@JLangr

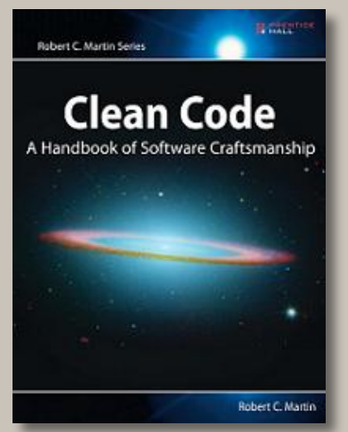












Workshop Overview

- Collaboration and software development
- Mobbing fundamentals
- Explanation of workshop
- Iterations 1 -> n
  - 20 minute break (~2:40)
- Wrap-up

We are uncovering **better ways of developing software**
by doing it and helping others do it.
Through this work we have come to value:

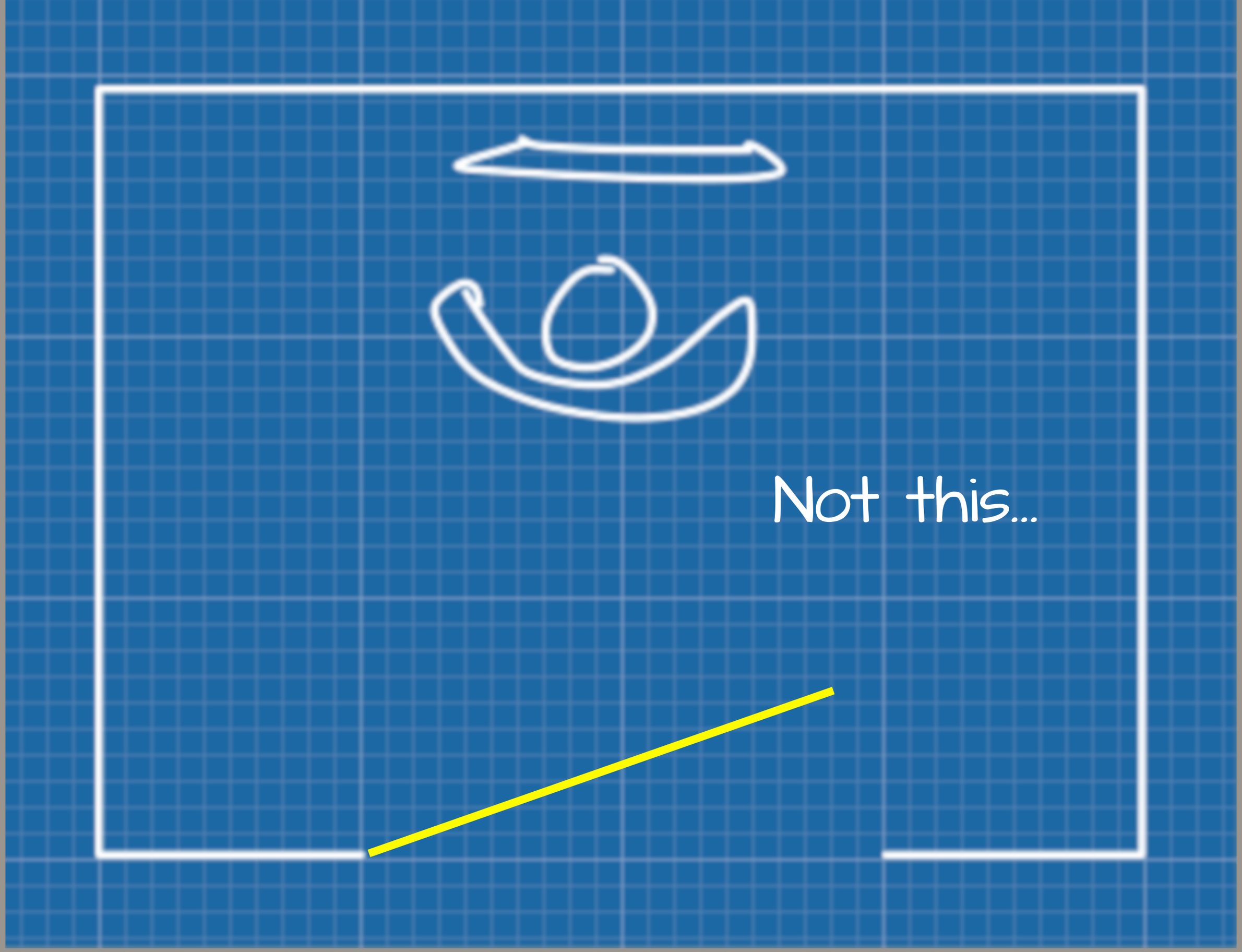
Individuals and interactions

over

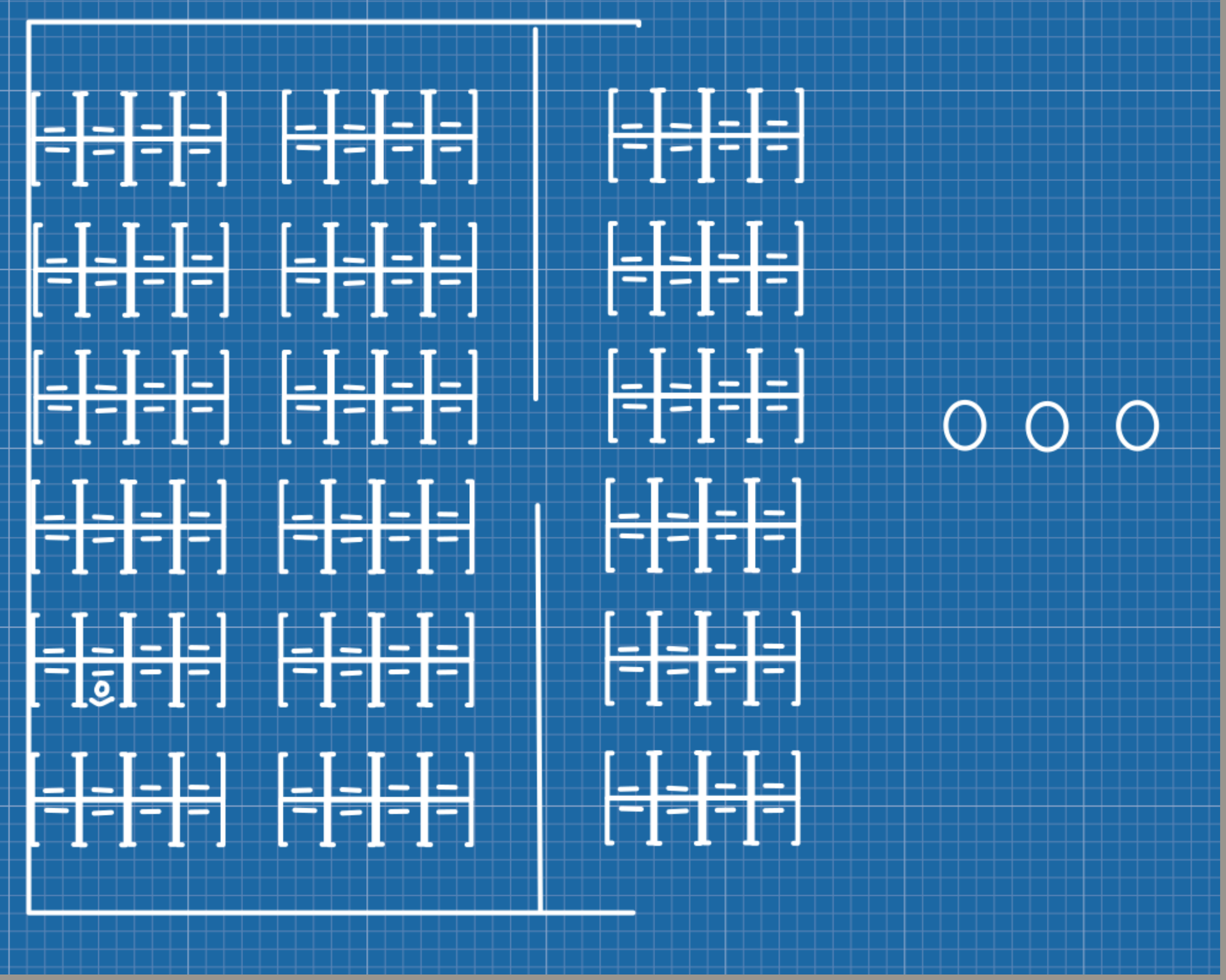
processes and tools

The **most efficient and effective** method of conveying information to & within a development team:

Face-to-face conversation



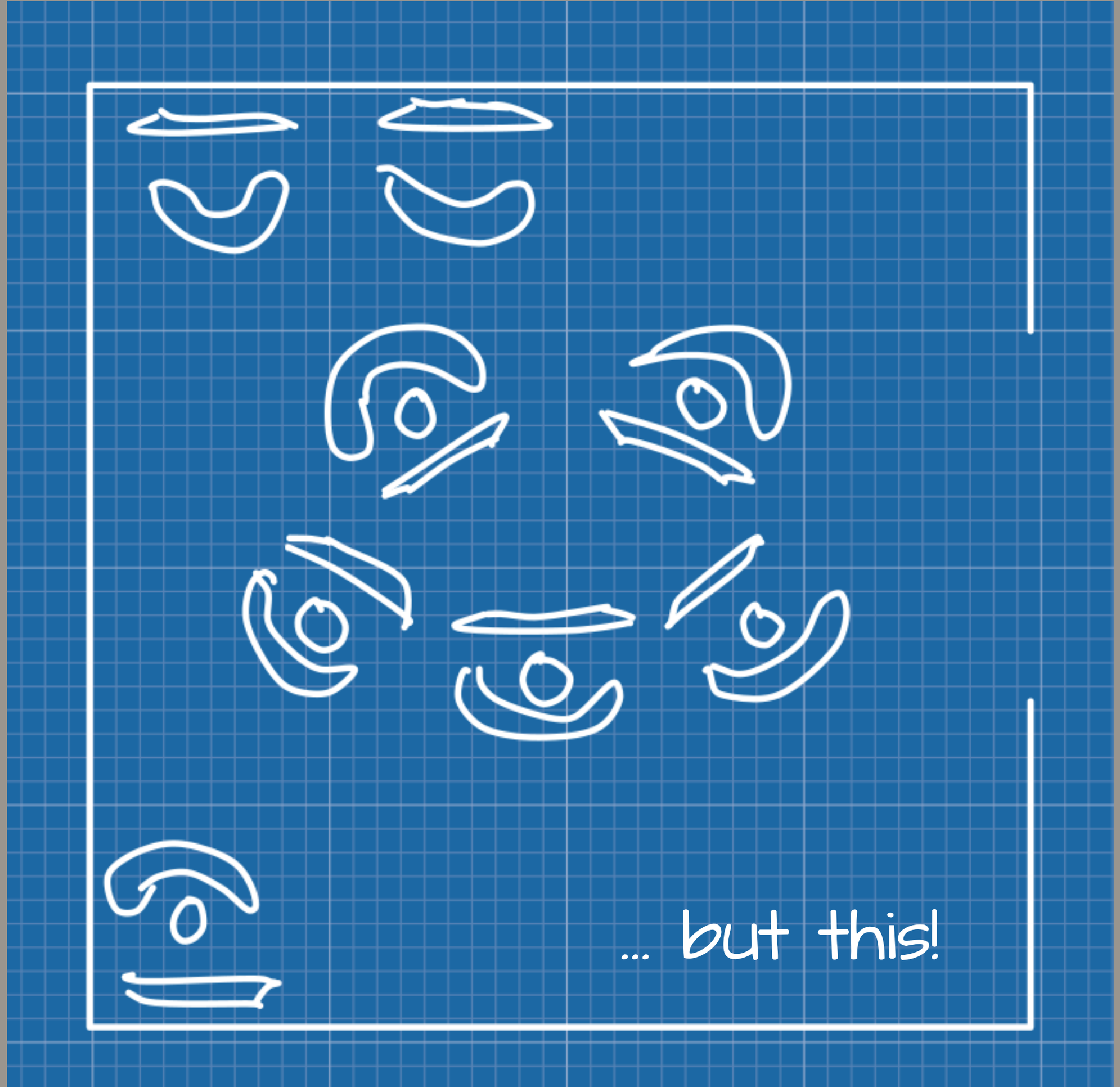
@JLangr



"CUBE FARM"

...or this...

@JLangr



@JLangr

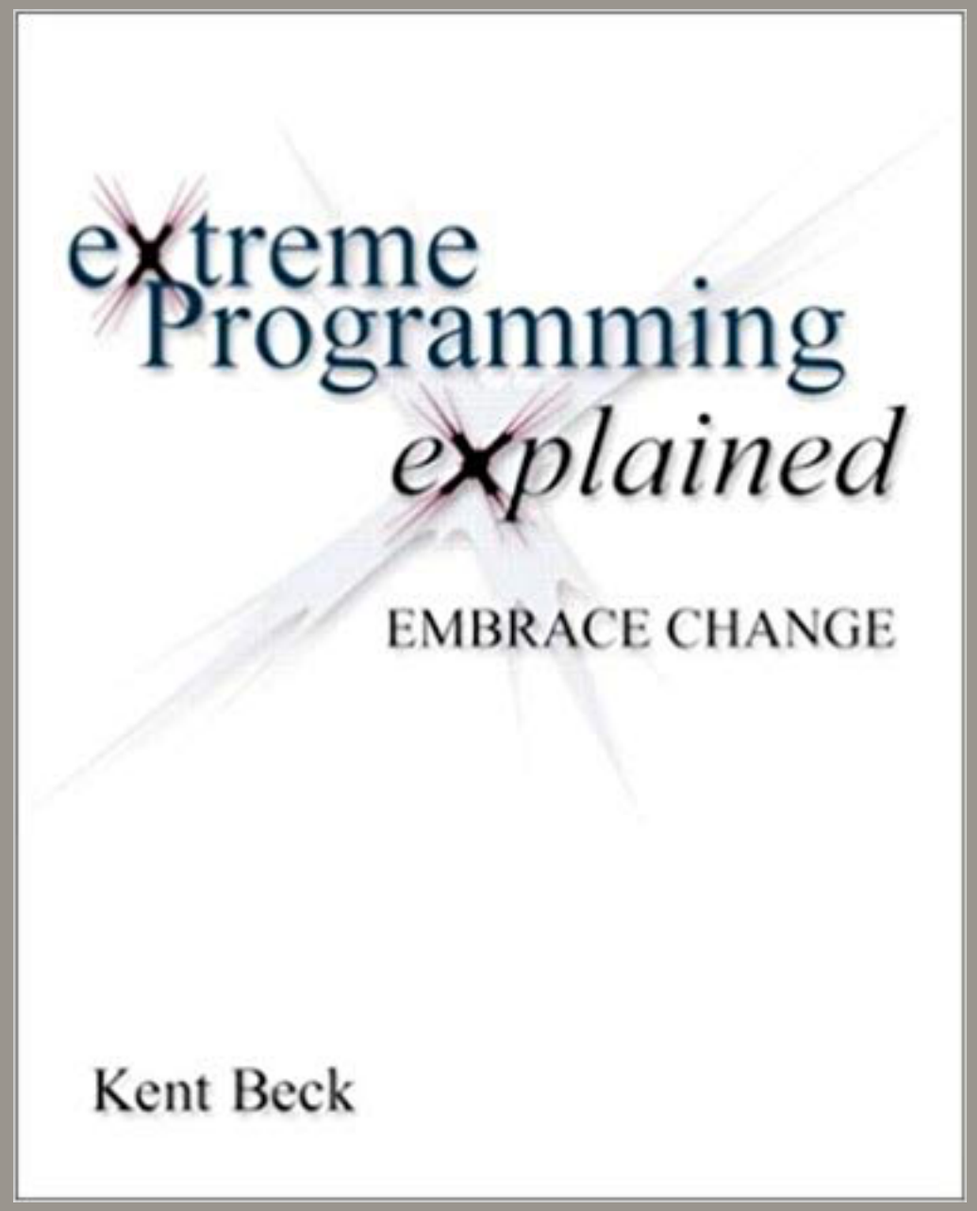
"Open Plan" Workspace

...And probably not this.

@JLangr

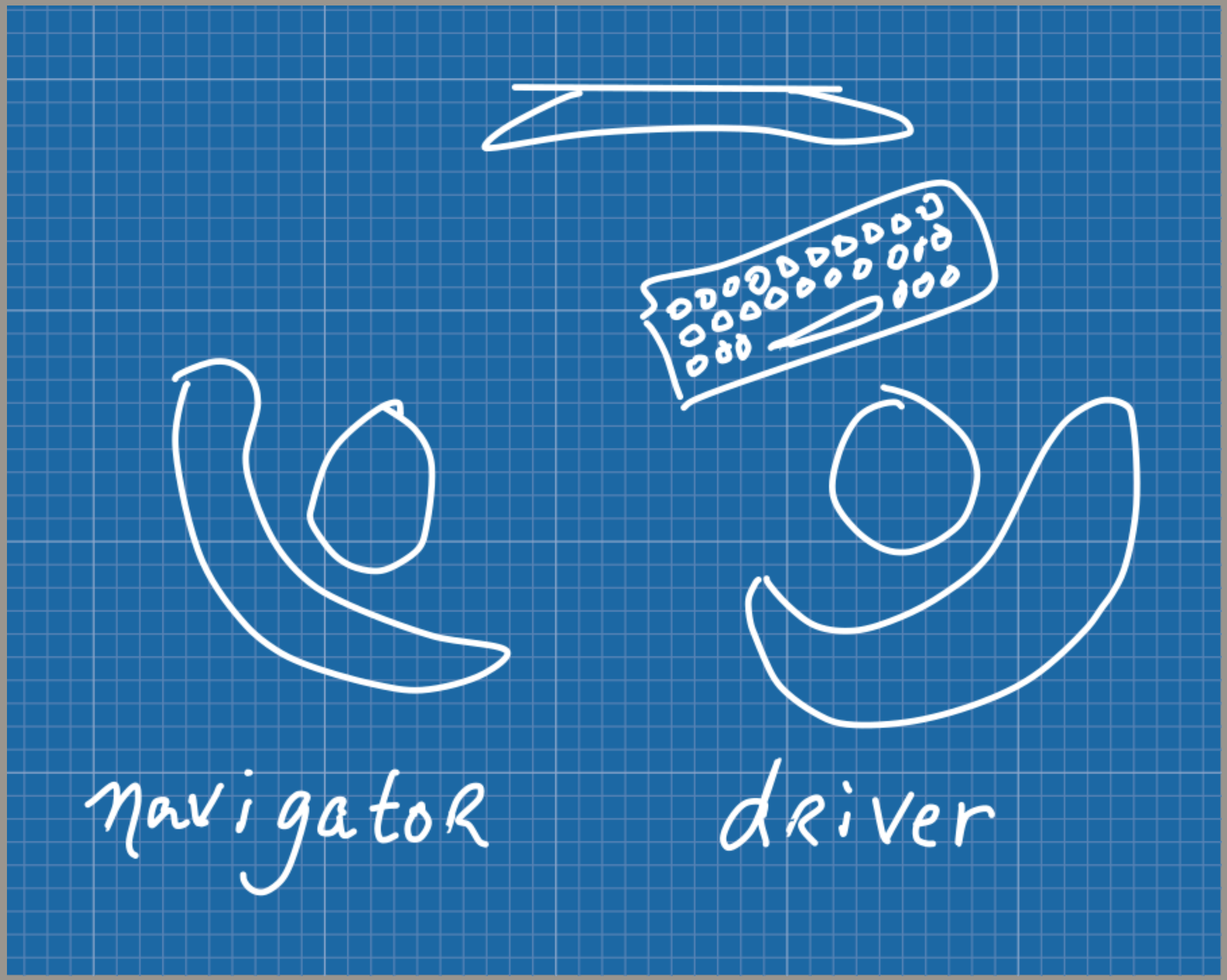
TRANSFORMING IDEA

1999



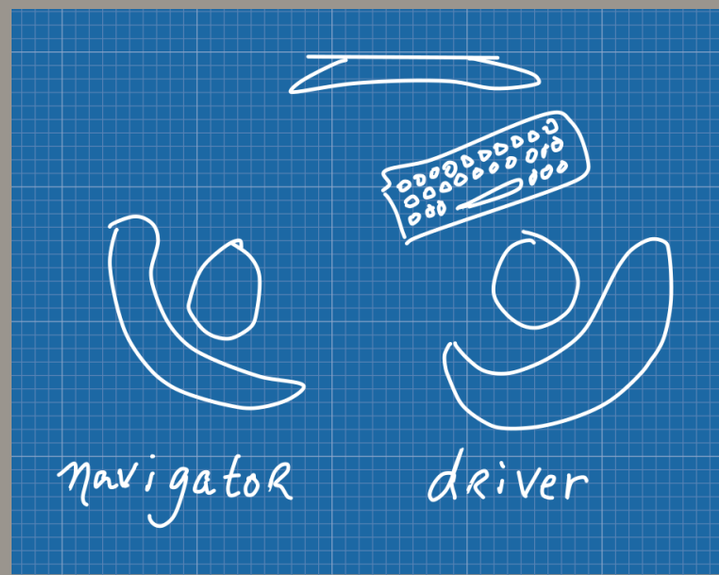
@JLangr

PAIR PROGRAMMING



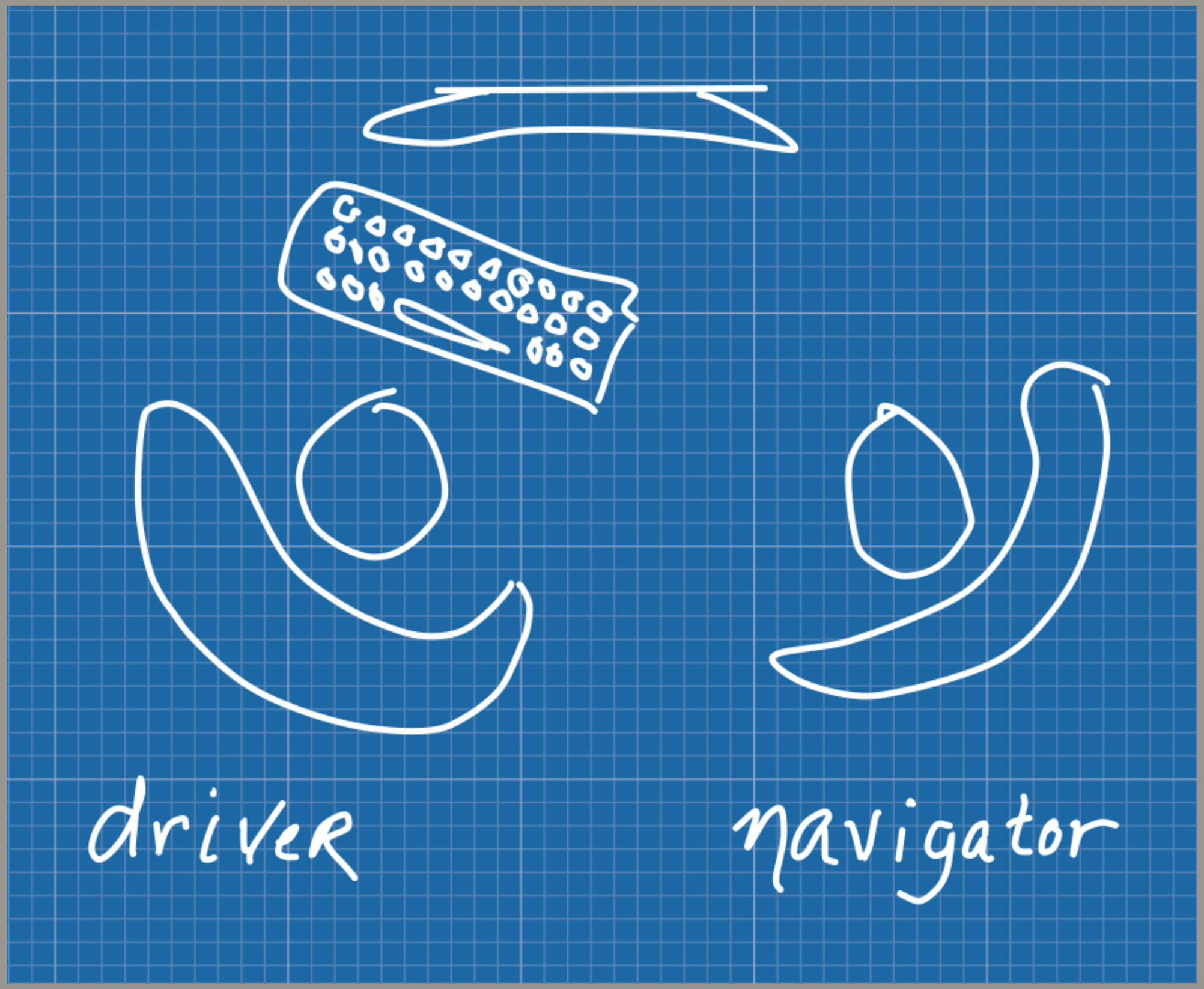
@JLangr

Why?

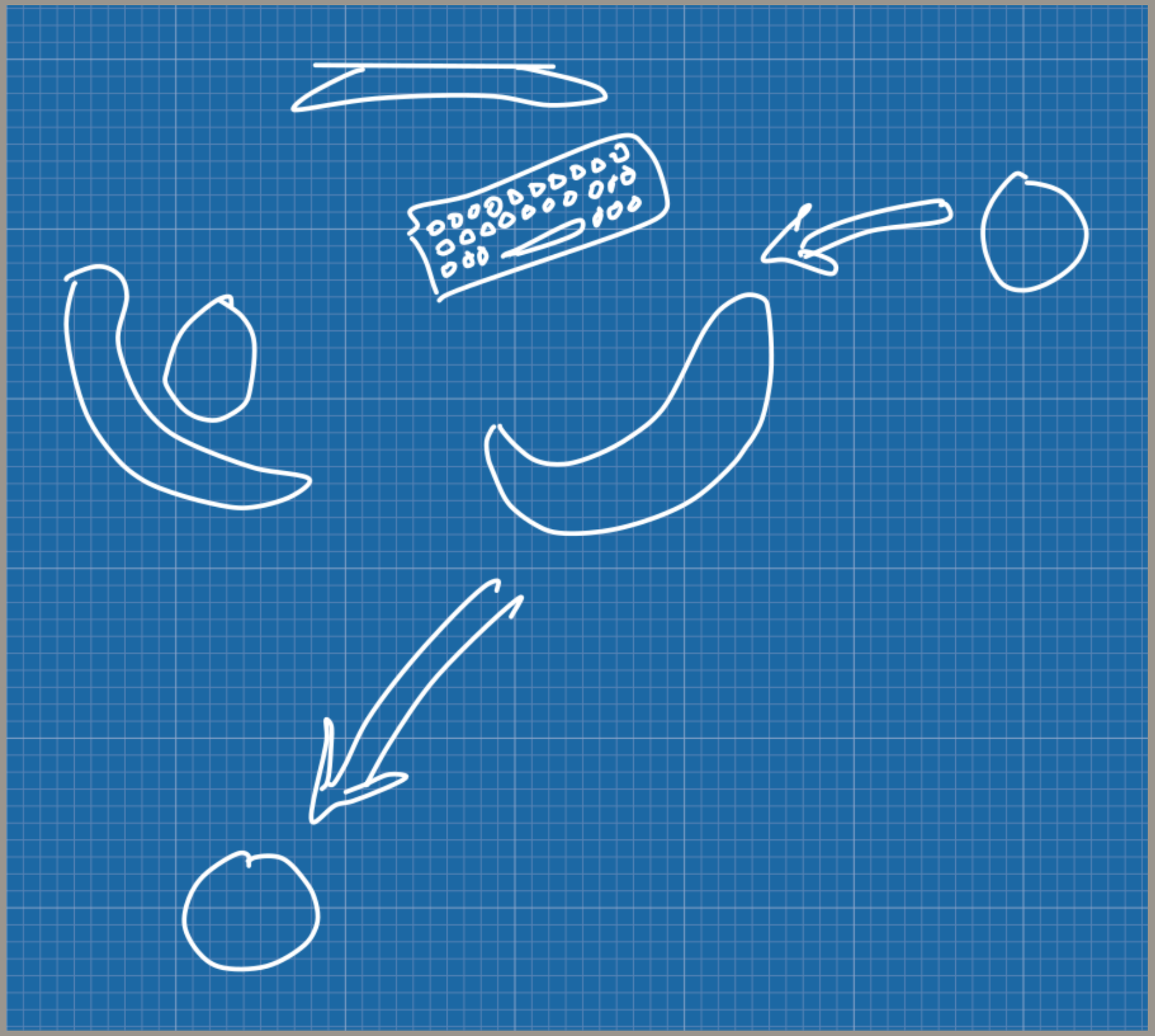


@JLangr

SWITCHING ROLES

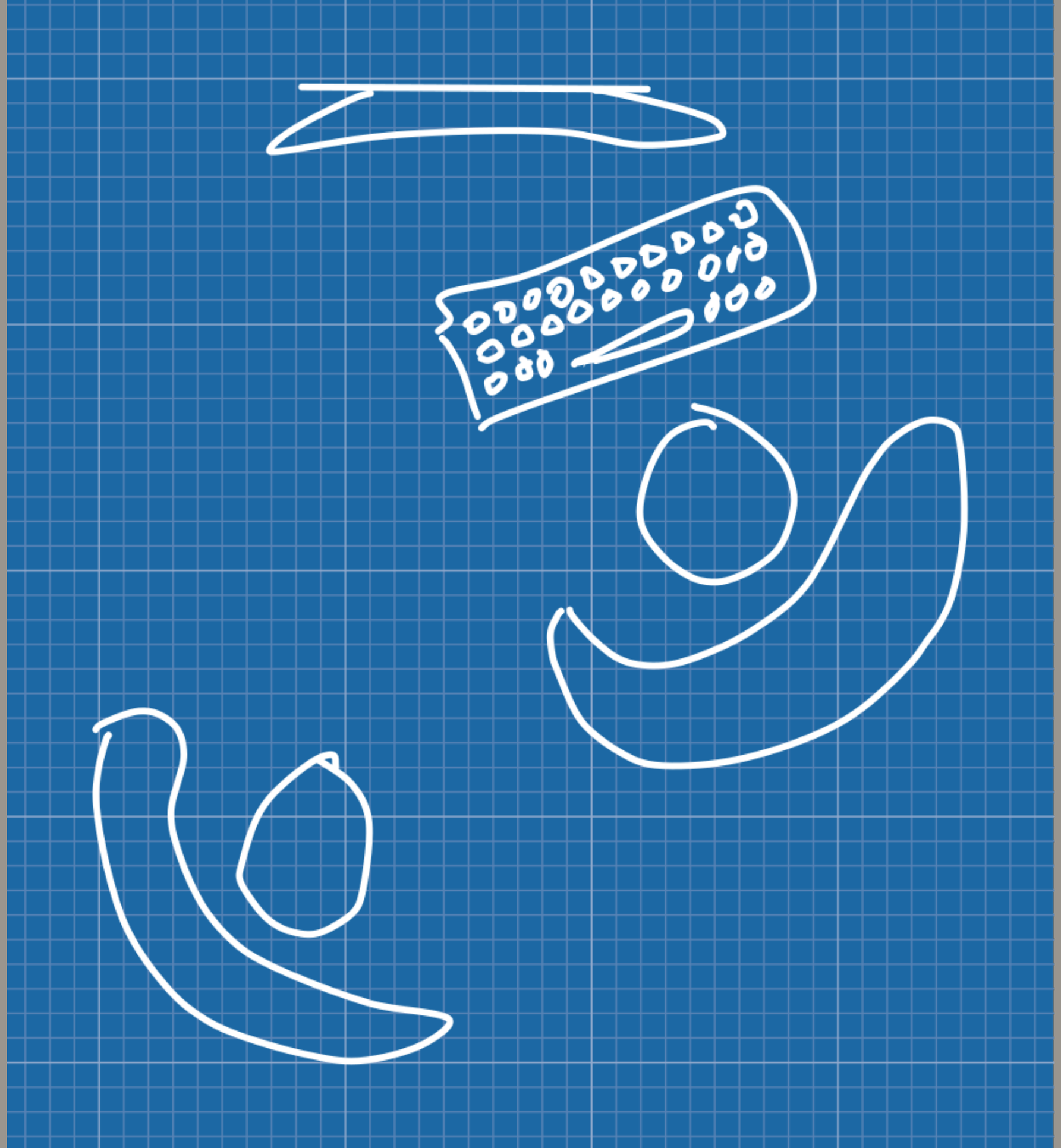


@JLangr



PAIR SWAPPING

@JLangr



NOT PAIRING!

@JLangr

SOME PAIRING CHALLENGES

- Fear
- Worker-Rester
- Domination
- Novice-novice
- Hygiene / health
- Interactions
- Physical accommodations
- Accessibility needs
- Shared time
- Individual resistance
- Preferred environments

@JLangr

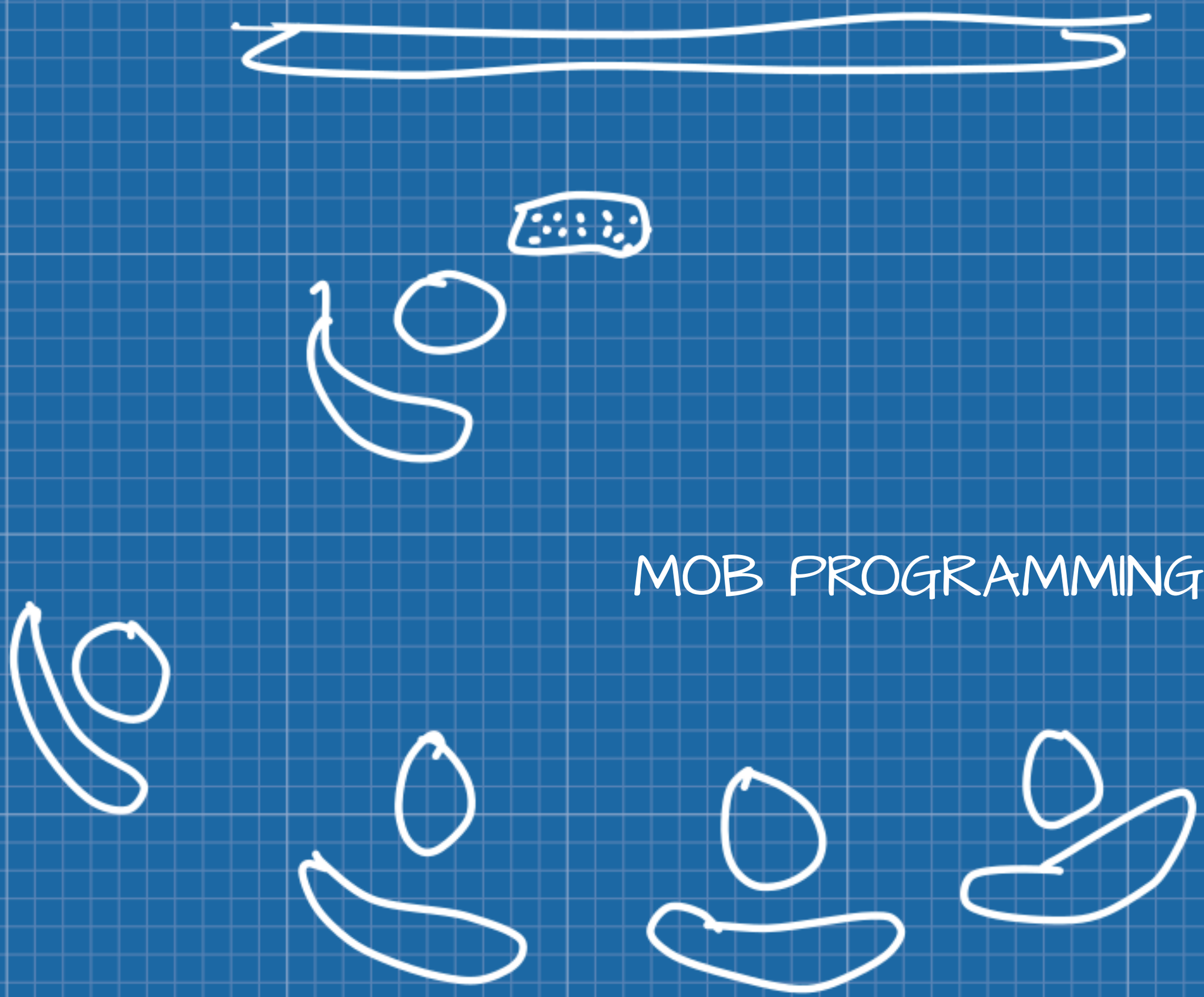
MOB PROGRAMMING

"All the brilliant people working on the same thing,
at the same time,
in the same place,
and on the same computer."

Photo courtesy Agile Ailliance.

-- Woody Zuill

http://mobprogramming.org



@JLangr

The Right Tool for the Job

2 Rules

+

Comfort

+

Ongoing Attention

@JLangr

Rule #1: NO THINKING AT KEYBOARD

"For an idea to go from your head into the computer it MUST go through someone else's hands."

-- Llewellyn Falco

http://llewellynfalco.blogspot.com/2014/06/llewellyns-strong-style-pairing.html

Rule #2: TIMED ROTATION

@JLangr

@JLangr

"FEAR," Courtesy Kevin B 3, https://www.flickr.com/photos/22850192@N03/8290599649/
License: https://creativecommons.org/licenses/by/2.0/

Comfort & Safety

Breaks

Comfortable environment

Safety

Parking lot

?

Ongoing Attention

Mob fatigue

New folks

Interviewing

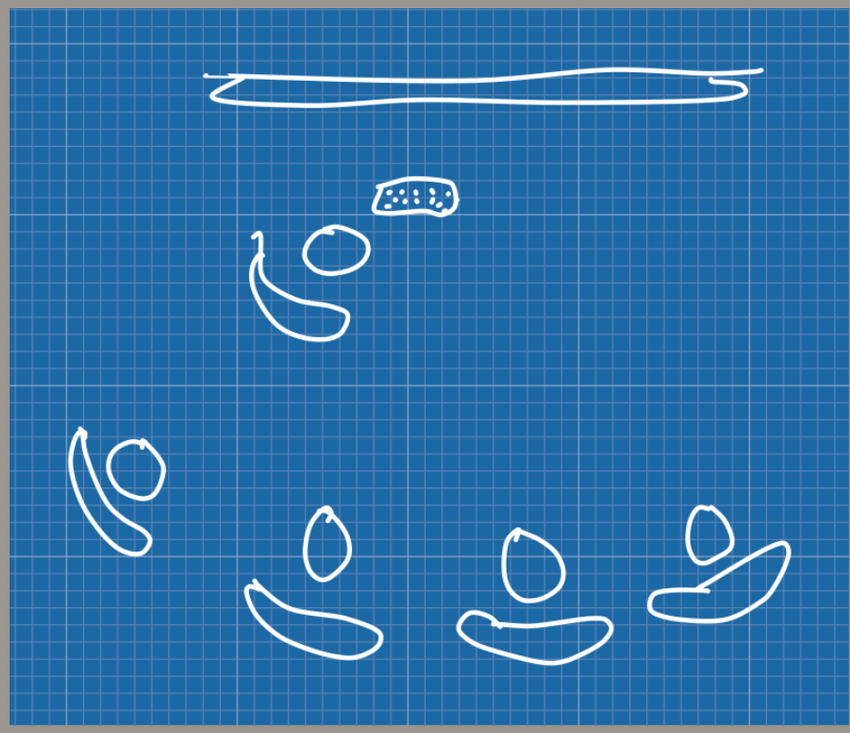
When / when not

Retrospecting

Interruptions

"We go faster."

Why?



@JLangr

The Workshop

What's like software development?

Q.

SongReducer.js — app

EXPLORER
OPEN EDITORS
APP
DeleteCommand.js
DeleteCommand.test.js
DuplicateNoteCommand.js
DuplicateNoteCommand.test.js
IncrementCommand.js
IncrementCommand.test.js
index.js
RebarCommand.js
RebarCommand.test.js
ToggleAccidentalCommand.js
ToggleAccidentalCommand.test.js
ToggleDotCommand.js
ToggleDotCommand.test.js
ToggleRestCommand.js
ToggleRestCommand.test.js
reducers
index.js
SongReducer.js
SongReducer.test.js
SynthReducer.js
SynthReducer.test.js
UIReducer.js
UIReducer.test.js
store
ui
Line.js
NoteWidget.js
TieWidget.js
util
Draw.js
Draw.test.js
UI.js
UI.test.js
Validations.js
Validations.test.js

SongReducer.test.js

```
361     expect(newState.song.tracks[0].flats).toEqual(['F4'])
362   })
363
364   it('removes when already exists', () => {
365     const state = { song: { name: 'x', tracks: [{flats: ['
366
367     const newState = SongReducer(state, actions.addFlat(0,
368
369     expect(newState.song.tracks[0].flats).toEqual([])
370   })
371 })
372
373 describe('trackData', () => {
374   it('represents the track given an id', () => {
375     const state = { song: { name: 'x', tracks: [{name: 'x'
376
377     const track = trackData(state, 1)
378
379     expect(track).toEqual({name: 'y'})
380   })
381 })
382
383 describe('answers whether or not track has notes in treble
384   it('does when any note is middle C or higher', () => {
385     const song = { tracks: [{ id: 0, notes: new NoteSequen
386     expect(hasTrebleNotes(song, 0)).toBeTruthy()
387   })
388
389   it('does not count rests', () => {
390     const noteSequence = new NoteSequence()
391     noteSequence.add(Note.Rest(Duration.quarter))
392
393     const song = { tracks: [{ id: 0, notes: noteSequence }
```

SongReducer.js

```
258         volume: defaultTrackVolume
259       }
260       return { ...state, song: { ...s
261     }
262
263     case type.REMOVE_SONG:
264     {
265       return { ...state, song: INITIA
266     }
267
268     case type.REPLACE_SONG:
269     {
270       const newSong = action.payload
271       newSong.tracks = newSong.tracks
272         const notes = track.notes.map
273         if (!track.volume) track.volu
274         return { ...track, notes: new
275       })
276       newSong.isDirty = false
277       return { ...state, song: newSon
278     }
279
280     case type.SET_VOLUME:
281     {
282       const { trackIndex, volume } =
283       return  updateState_setVolume(s
284     }
285
286     case type.SONG_LIST:
287     {
288       return { ...state, songList: co
289     }
290
```

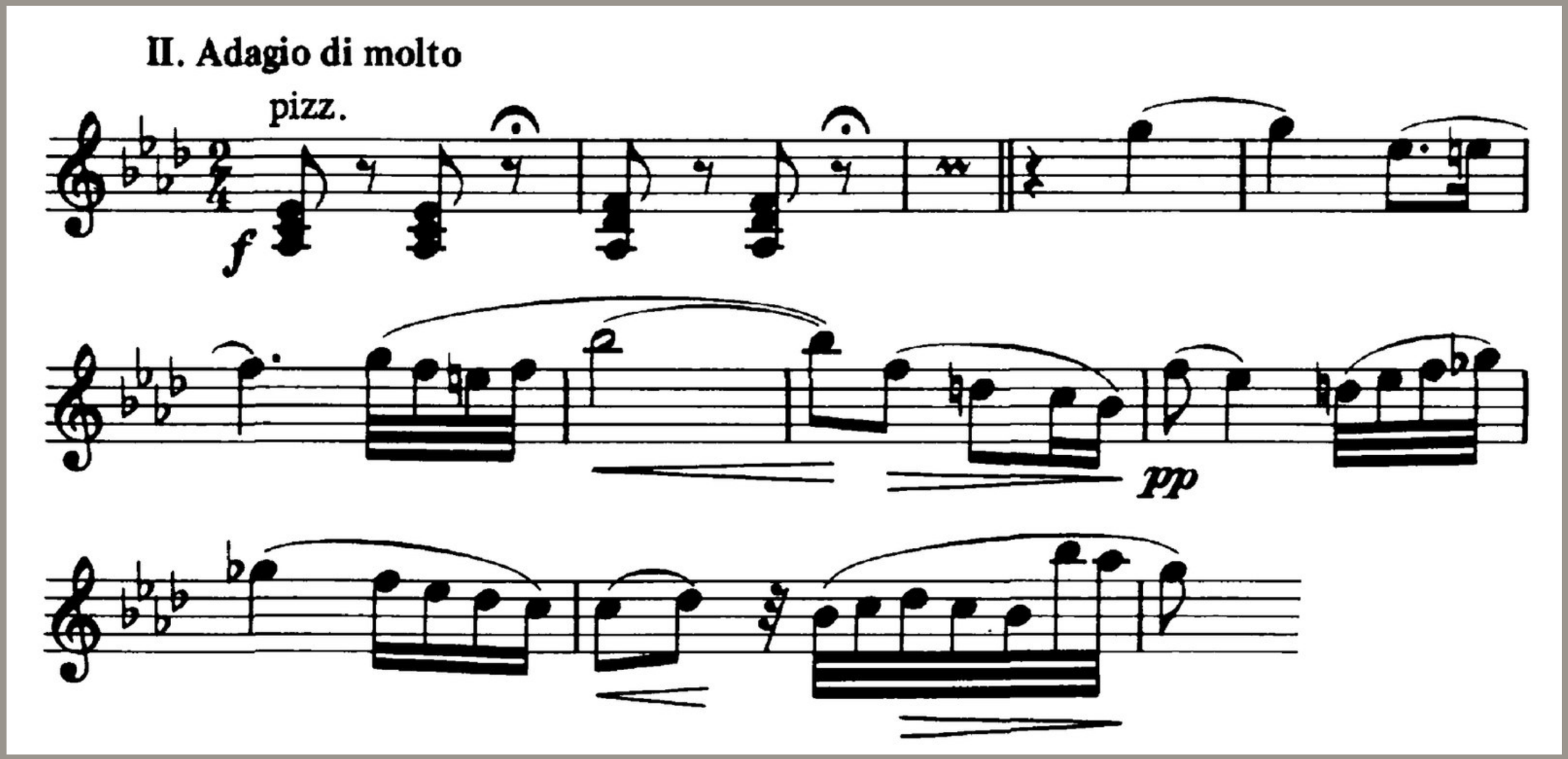
master* 0 0 Live Share

Key characteristics of software development?

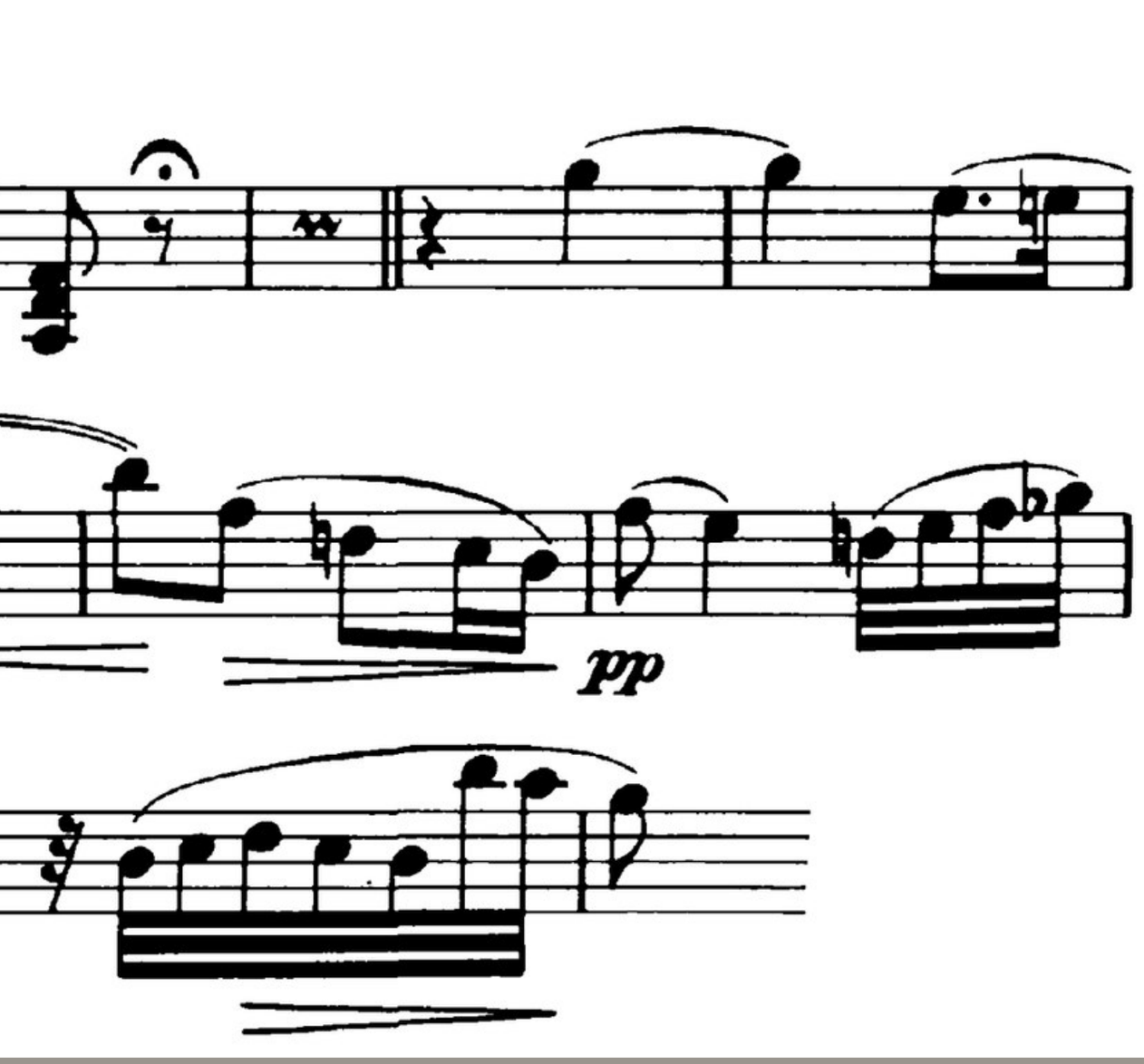
Key challenges in software development?

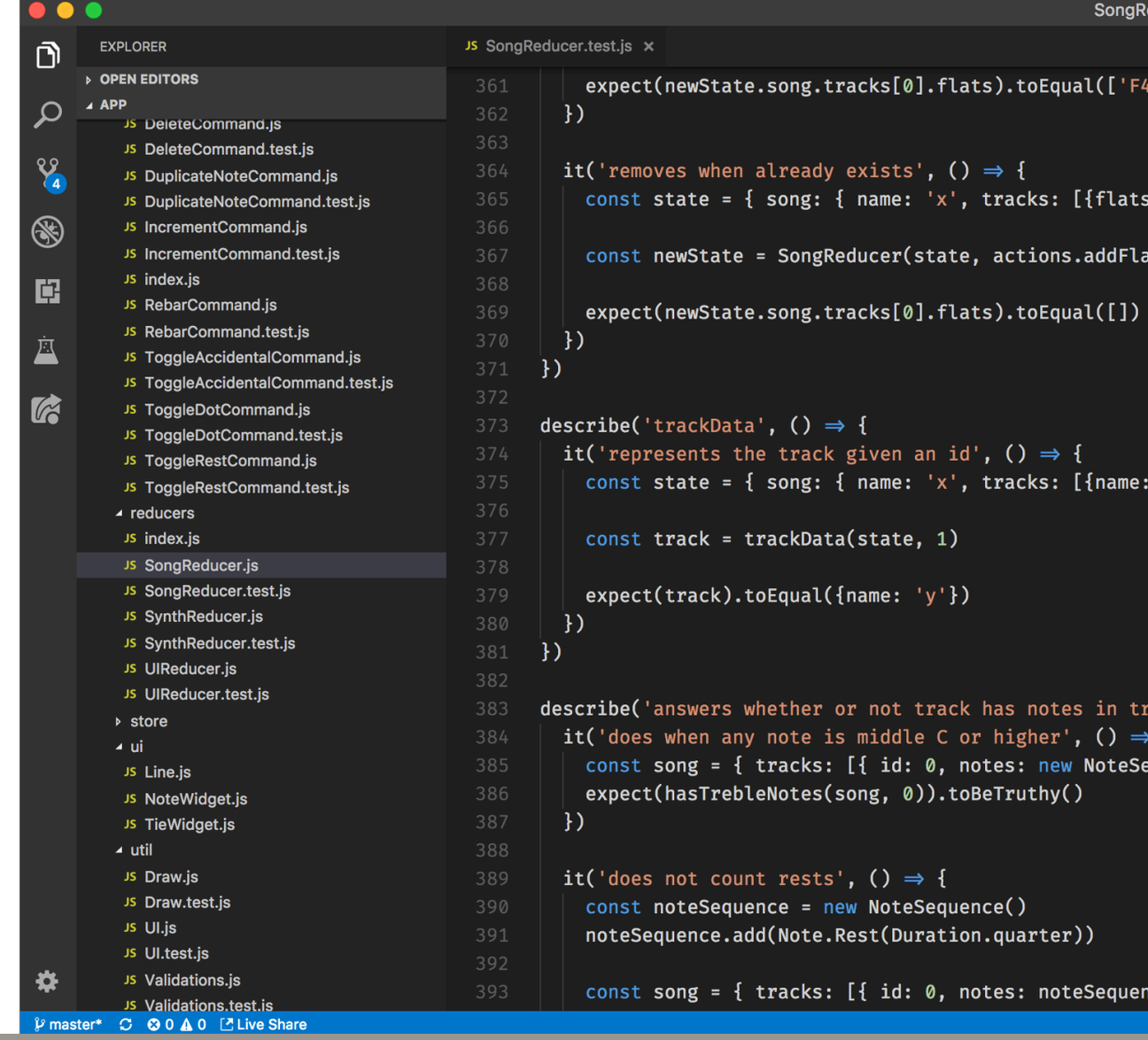
@JLangr

How is composing *similar to* software development?



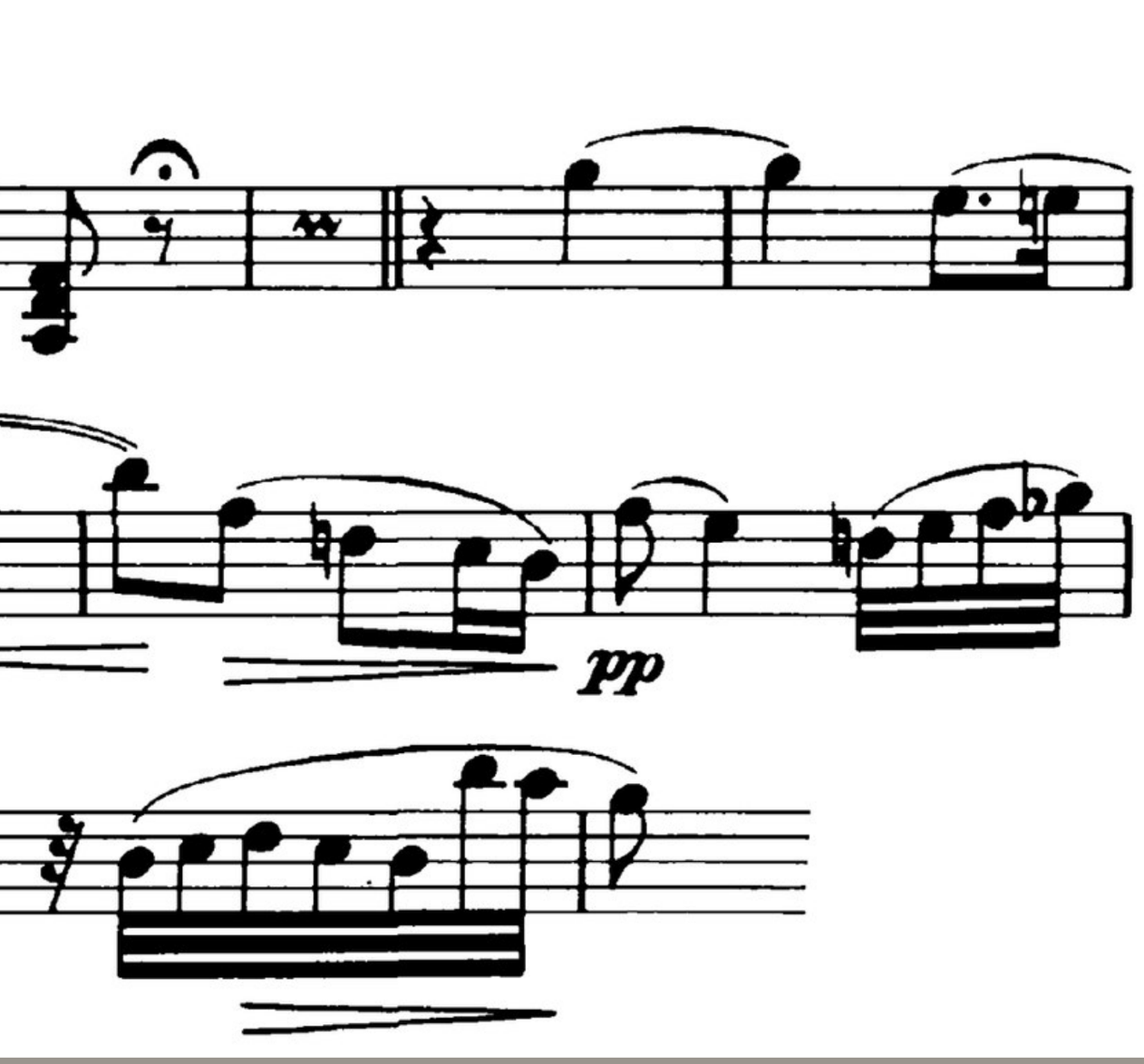
@JLangr

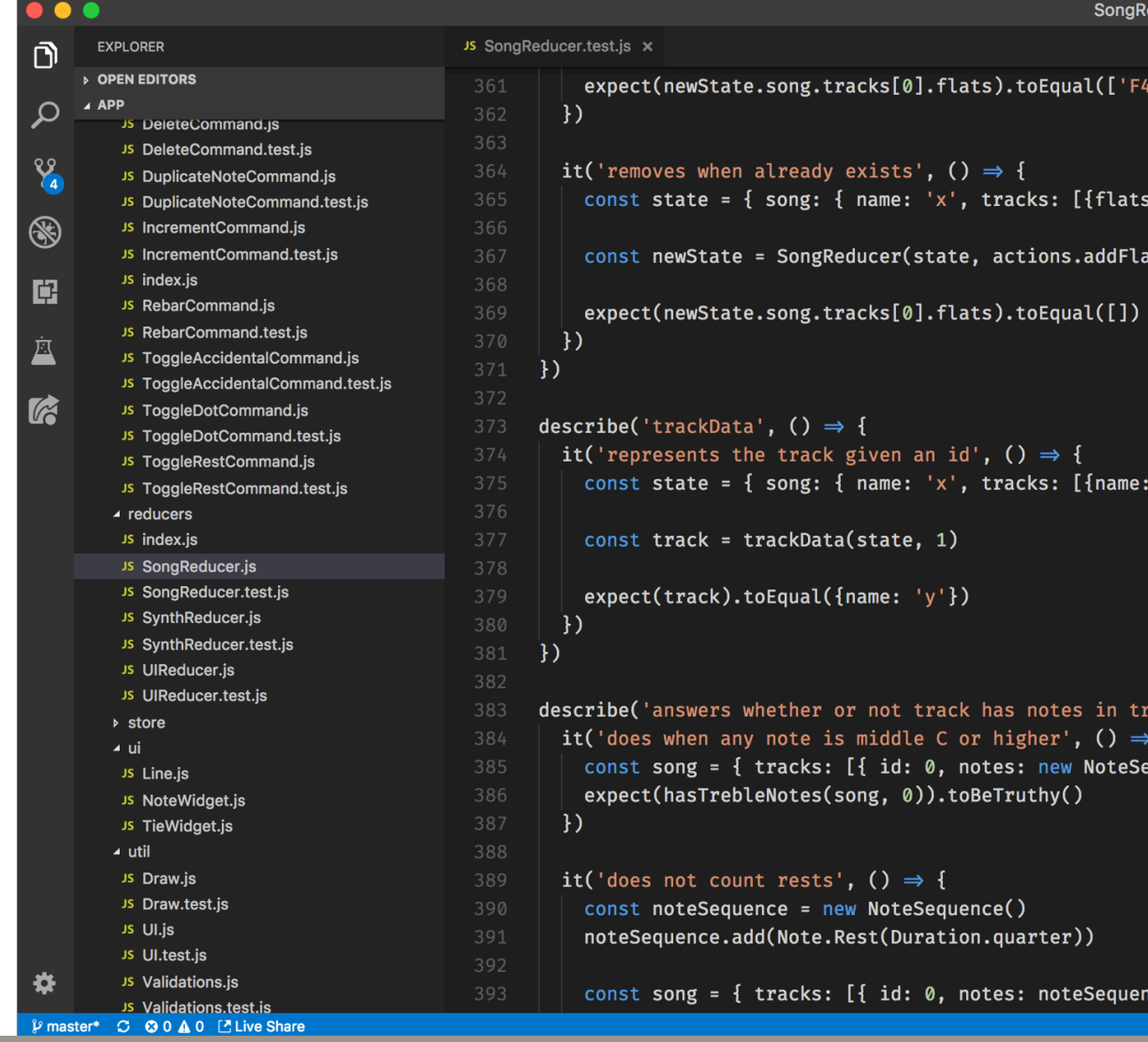




How is it *different from* software development?

@JLangr





Team or individual activities?

@JLangr

What Do We Know About Music?

@JLangr

https://www.pexels.com/photo/people-coffee-meeting-team-7096/

Translation Levels

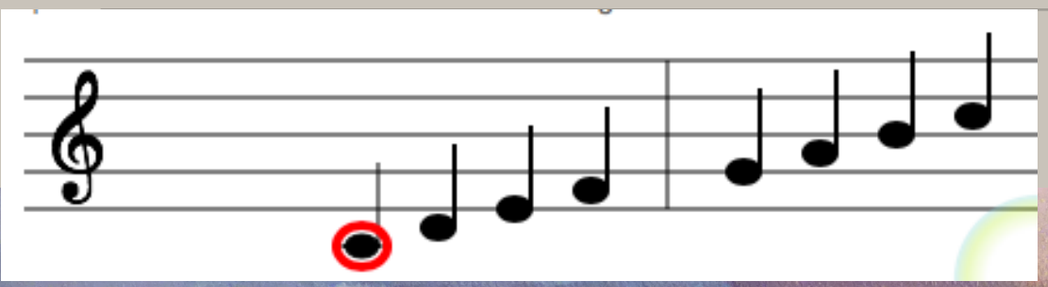
An Iteration

- Overview of next mobbing objective / rule
  - What to look for / think about
- Overview of task
- ~20 minutes of development time
- Demos
- Review of objective
- Mini-retro

Iterations

- Installing Geektone
- Exploring Geektone (twinkle twinkle)
- Melody
- Chorus
- Tension / release
- Rhythm and "space"
- Counter-melody
- Intro
- Outro

Geektone & Music Basics



Notes

- Duration

1/4	1/2	1	1/8	1/16
quarter	half	whole	eighth	sixteenth

- Dotted

One-and-a-half times normal duration

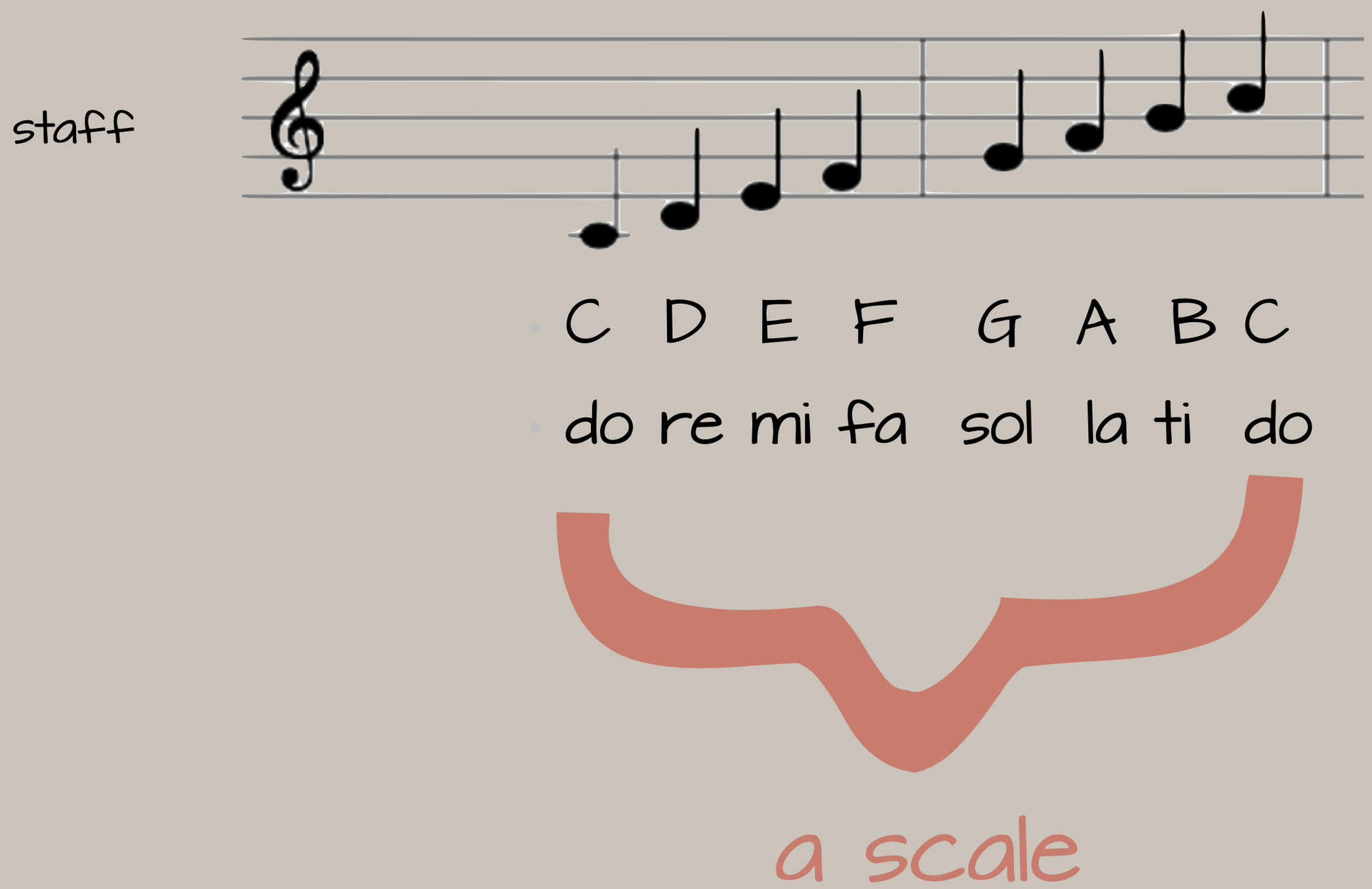
@JLangr

Rests

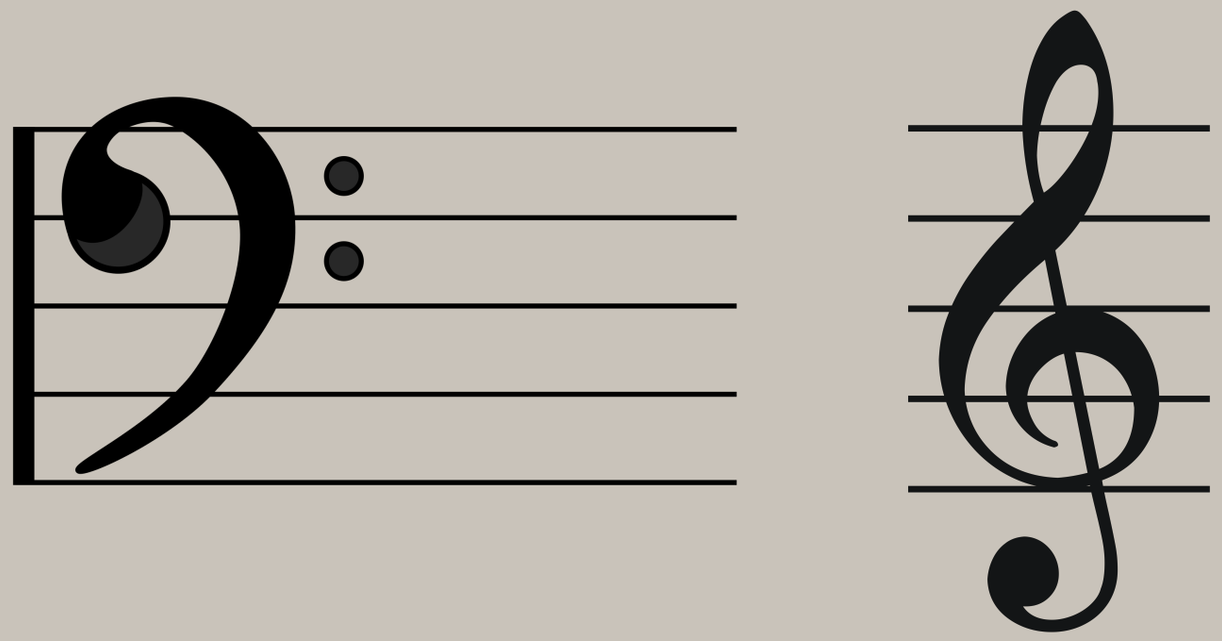
- Duration

1/4	1/2	1	1/8	1/16
quarter	half	whole	eighth	sixteenth

Notes Have Pitch



@JLangr



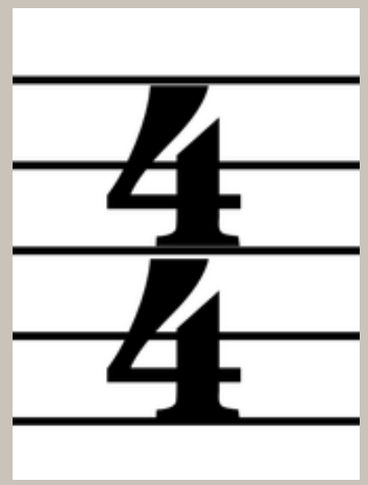
Bass & Treble Clefs

@JLangr

Sharps & Flats

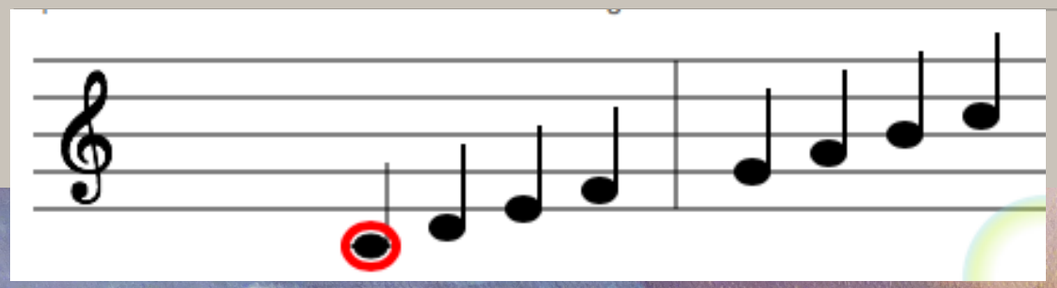
- ♯ Sharp: half step up (semitone)
- ♭ Flat: half step down
- Applied to all notes of same pitch!

Measures & Beats



- 4/4 time:
  - 4 beats per measure
  - quarter note gets the beat

Composing Challenge



The Challenges

- 45 or 60 seconds
- Television commercial
  - Feel-good product ad, emotive charity ad, medical malady ad, ...
- Movie title track
  - Horror, comedy, drama, adventure, ...
- Television montage
  - Uplifting; increasingly desperate; tear-jerking, ...

- Introduce yourselves (5 min)
- Choose challenge (5 min)
- Discuss theme (5 min)
  - Agree on the core emotion(s) to provoke
    - "anxiety," "joy," "relaxation"
  - Agree on a few of goals
    - "punchy," "meandering," "increasingly frantic"

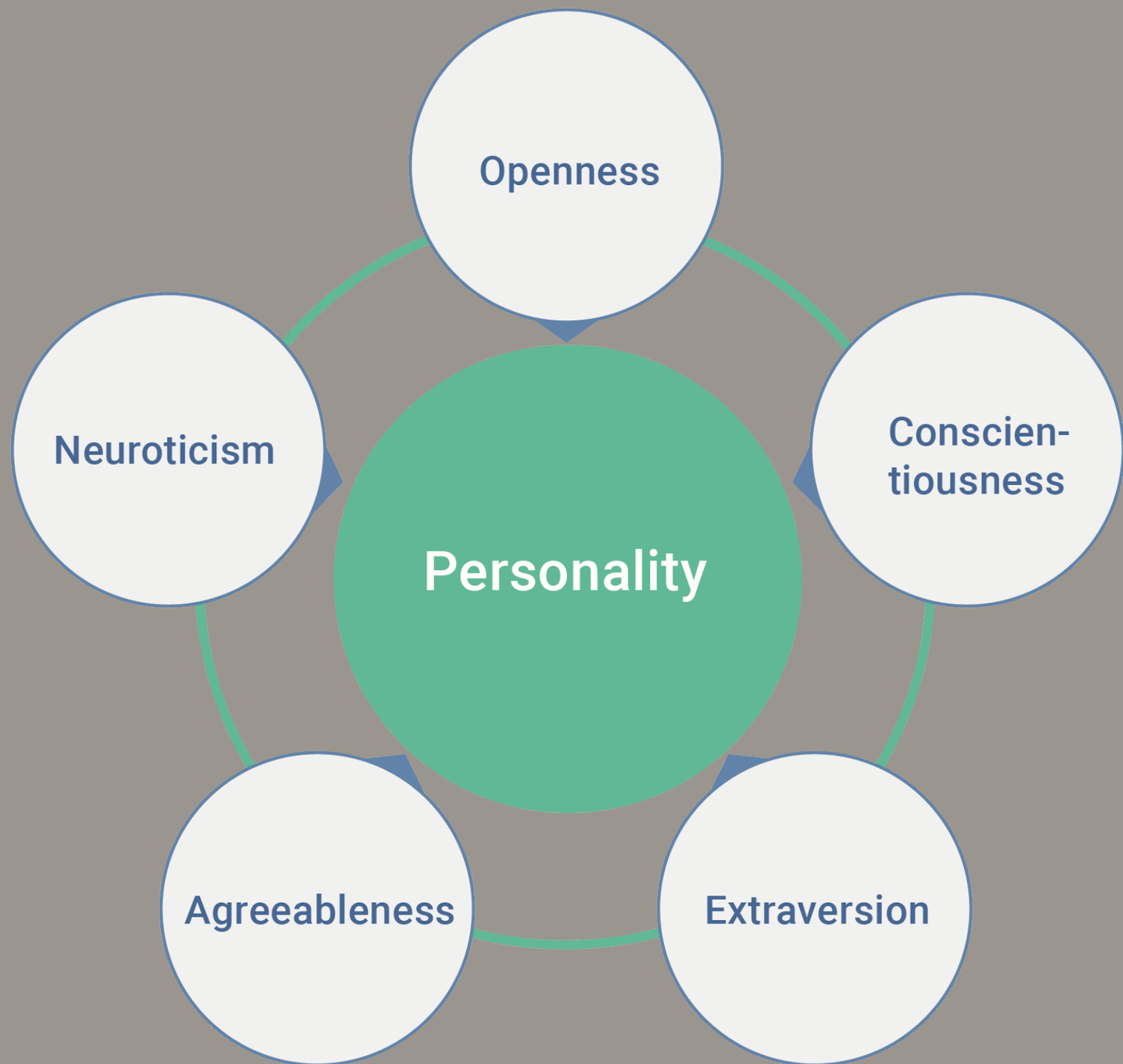
Mob !

Review.

@JLangr

Reflect.

@JLangr



@JLangr

https://en.wikipedia.org/wiki/Big_Five_personality_traits

"Pomodoro Kitchen Timer for Activity Logging," courtesy Andy Roberts
https://www.flickr.com/photos/aroberts/6332067642
https://creativecommons.org/licenses/by/2.0/
35
30
Pomodoros
@JLangr
https://en.wikipedia.org/wiki/Pomodoro_Technique

Song Structure

Intro	Verse	Chorus	Verse	Chorus	Bridge	Verse	Chorus	Outro

@JLangr

Basic Melody

(A) common thread through a song

- Range: An octave and a half
- Perhaps contains some repeating elements
- Mostly "stepwise" with occasional leaps
- Climactic moment
  - Highest notes?

Verse

- Tells a story -- often new lyrics
- Details of the song
- Supports the chorus

Intro	Verse	Chorus	Verse	Chorus	Bridge	Verse	Chorus	Outro

@JLangr

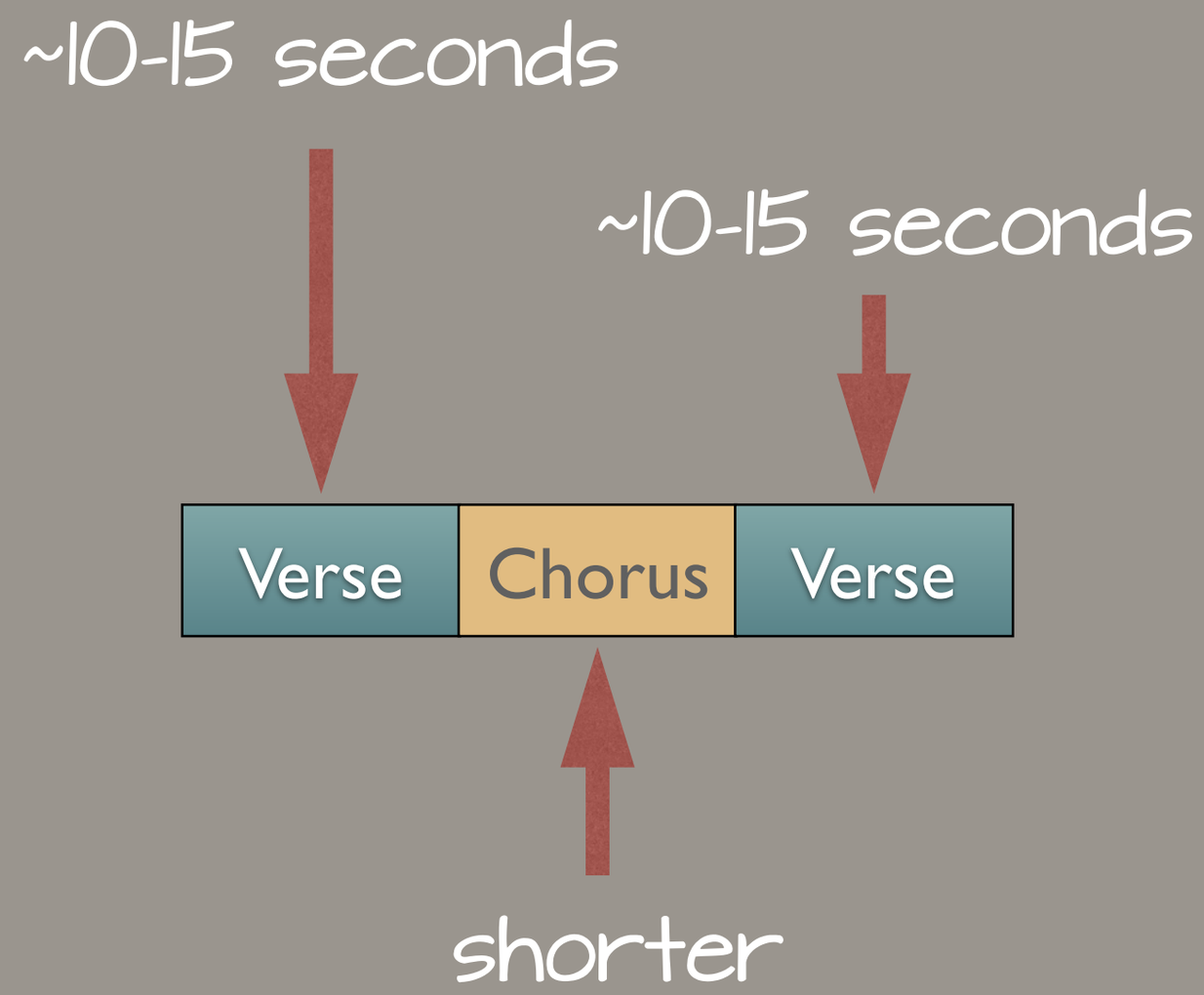
Chorus

- Repeated verse / phrase: a motif

Intro	Verse	Chorus	Verse	Chorus	Bridge	Verse	Chorus	Outro

- Short and simple!
- Repetition + simple variants
- Punctuate
  - Volume
  - Silence
  - Add'l instruments

Pomodoro: Compose a basic song



@JLangr

Rules Reminder

Five minute switch

Keyboardist drives only

Mob !

@JLangr

https://www.pexels.com/photo/chairs-device-electronics-eyeglasses-1181376/

Review.

@JLangr

Reflect.

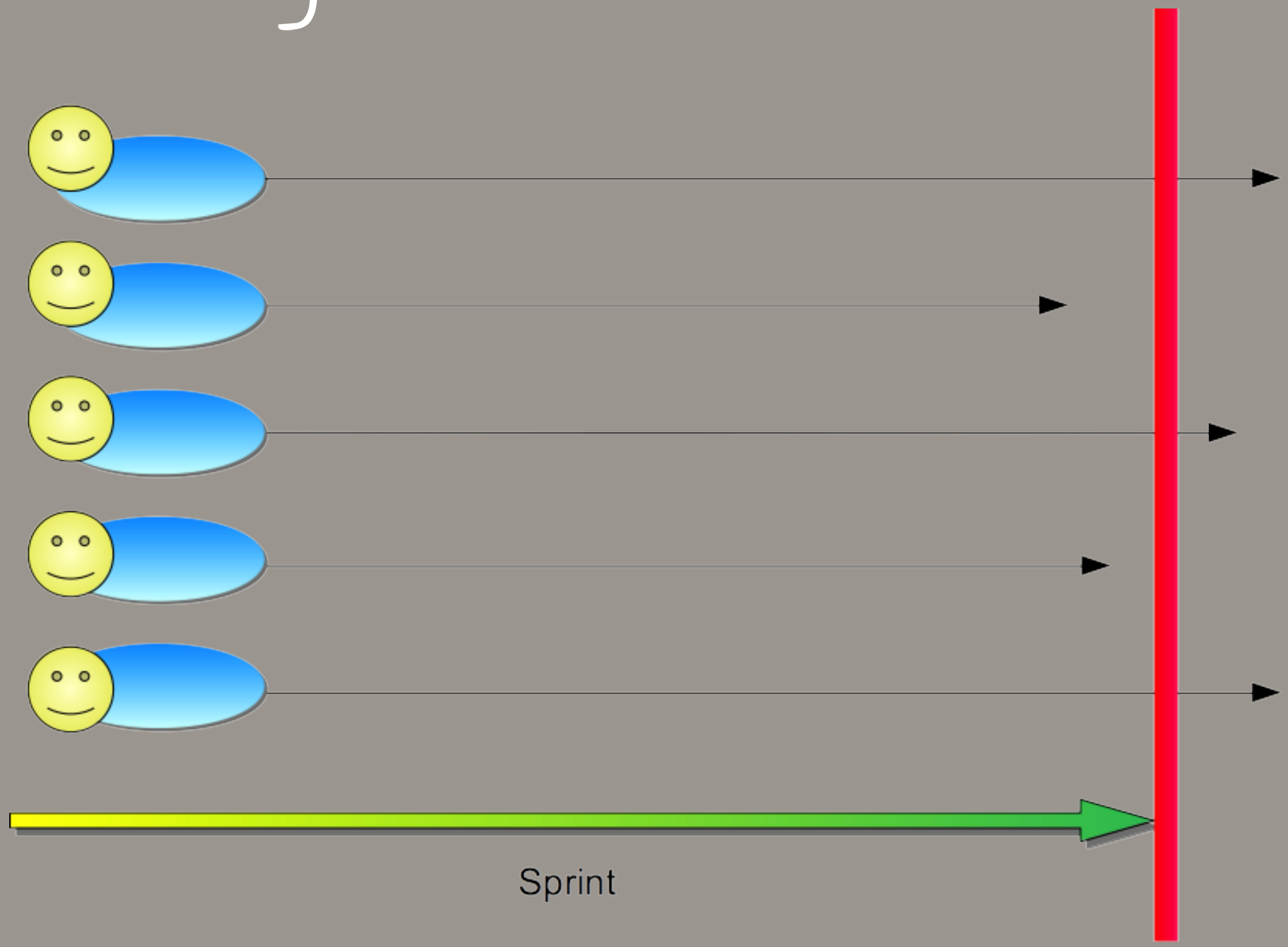
@JLangr

Flow

@JLangr

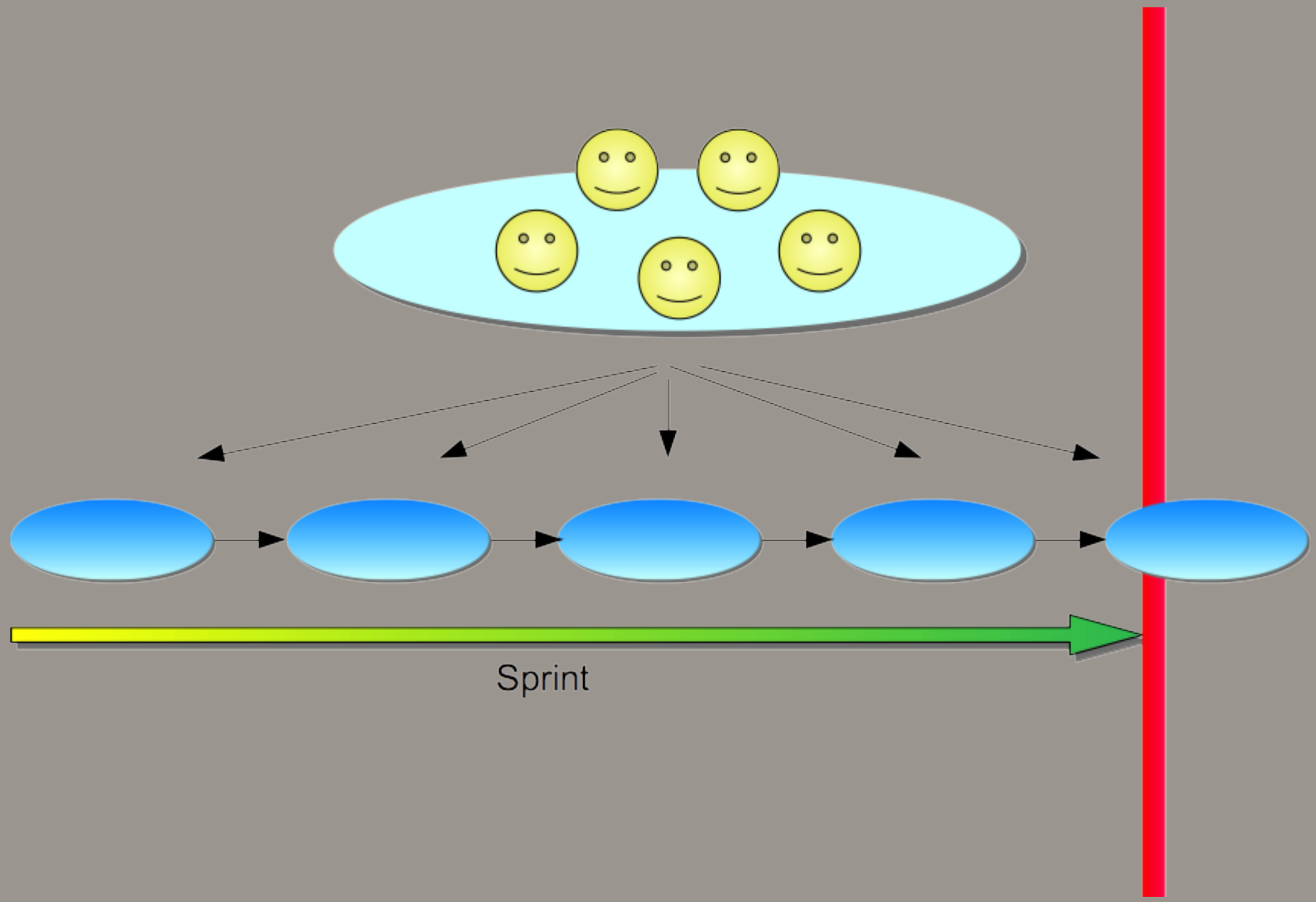
https://www.pexels.com/photo/architecture-auto-automobiles-bridge-210182/

Classic Agile Iteration



@JLangr

Reduced WIP



@JLangr

Creating "Space"

- Silence is golden
- Rests can:
  - add interest to melodies
  - create interesting rhythms
  - create suspense

Rhythm

- Brings character to the music
- Keeps the song together
- Makes us want to move

- Listen to some basic rhythms
- Start simple: be repetitive
- Tweak it!
- Clap it out

Pomodoro: Rhythm & Space

- Some iterate, some increment
- Seek to introduce space in the verse / chorus
- Create a new track for rhythm

Rules Reminder

Five minute switch

Keyboardist drives only

Mob !

@JLangr

https://www.pexels.com/photo/chairs-device-electronics-eyeglasses-1181376/

Review.

@JLangr

Reflect.

@JLangr

Tension / release

- Building unrest / conflict
- Needs release!
- Example: Chord progressions

- Rhythmic tension
- Melodic tension
- Bridge tension
- Instrumental tension
- Dynamic tension

Pomodoro: Introduce tension

@JLangr

Rules Reminder

Five minute switch

Keyboardist drives only

Mob !

@JLangr

https://www.pexels.com/photo/chairs-device-electronics-eyeglasses-1181376/

Review.

@JLangr

Reflect.

@JLangr

Additional Elements for Pomodoros

Counter-Melody

- Subordinate to main melody
  - Uses same scale
- Sounds good alone
- Should oppose, not overlap

Intro

- Grab the listener's attention
- Establish style / mood

Intro	Verse	Chorus	Verse	Chorus	Bridge	Verse	Chorus	Outro

- Perhaps use materials from rest of song
- Contrast:
  - Stripped down
  - Slower / faster
  - Grabby hook!...

Outro / Conclusion

- Inverse of intro?
- Wrap things up

Intro	Verse	Chorus	Verse	Chorus	Bridge	Verse	Chorus	Outro

- No rules
- Simple, might repeat elements of chorus / verse

Bridge

- Optional transition
- Pause / reflect on earlier parts of song
- Prepare listener for climax

Intro	Verse	Chorus	Verse	Chorus	Bridge	Verse	Chorus	Outro

- Opposing melody, chord progression
- Deepen emotional impact
- Connect smoothly

Wrapping Up

Some Tips

Law of two feet

Established core hours

Regular breaks!

Be there.

Why Faster

- Important discussions happen before code is integrated
  - Before it's too late
- Fewer / simpler rituals (standups, merging, defect tracking, etc)
- WIP of one--focus on delivery
- All the info in the room
- All the wisdom
- Flow
- Everyone understands the code
  - And why / decisions what went into it
- Consistent solutions
- Adherence to quality standards more likely

Things to Try

Guests

Warm-ups / katas
Speed challenges

Oblique strategies

http://www.rtqe.net/ObliqueStrategies/

@JLangr

"Kata," courtesy https://www.flickr.com/photos/superwebdeveloper/
License: https://creativecommons.org/licenses/by/2.0/

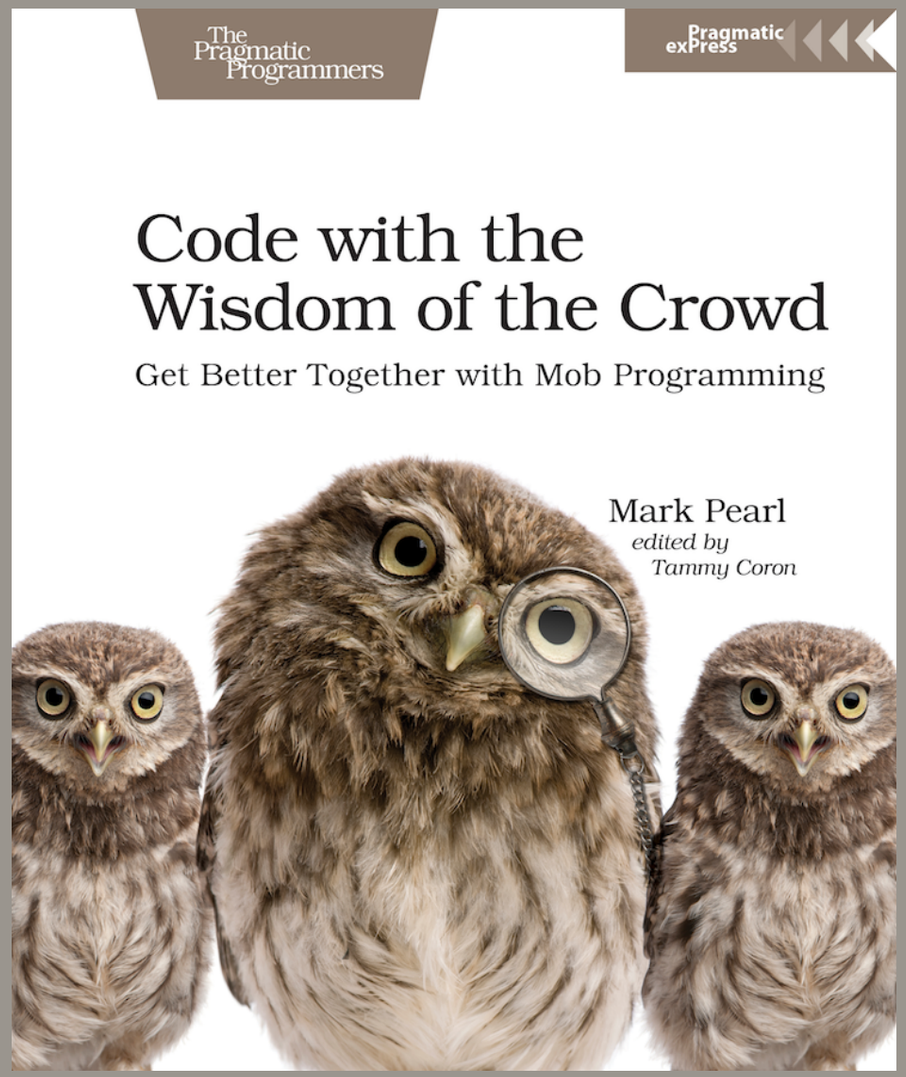
"Beautiful and dusty," courtesy https://www.flickr.com/photos/80435089@N08/
License: https://creativecommons.org/licenses/by/2.0/

Some Resources

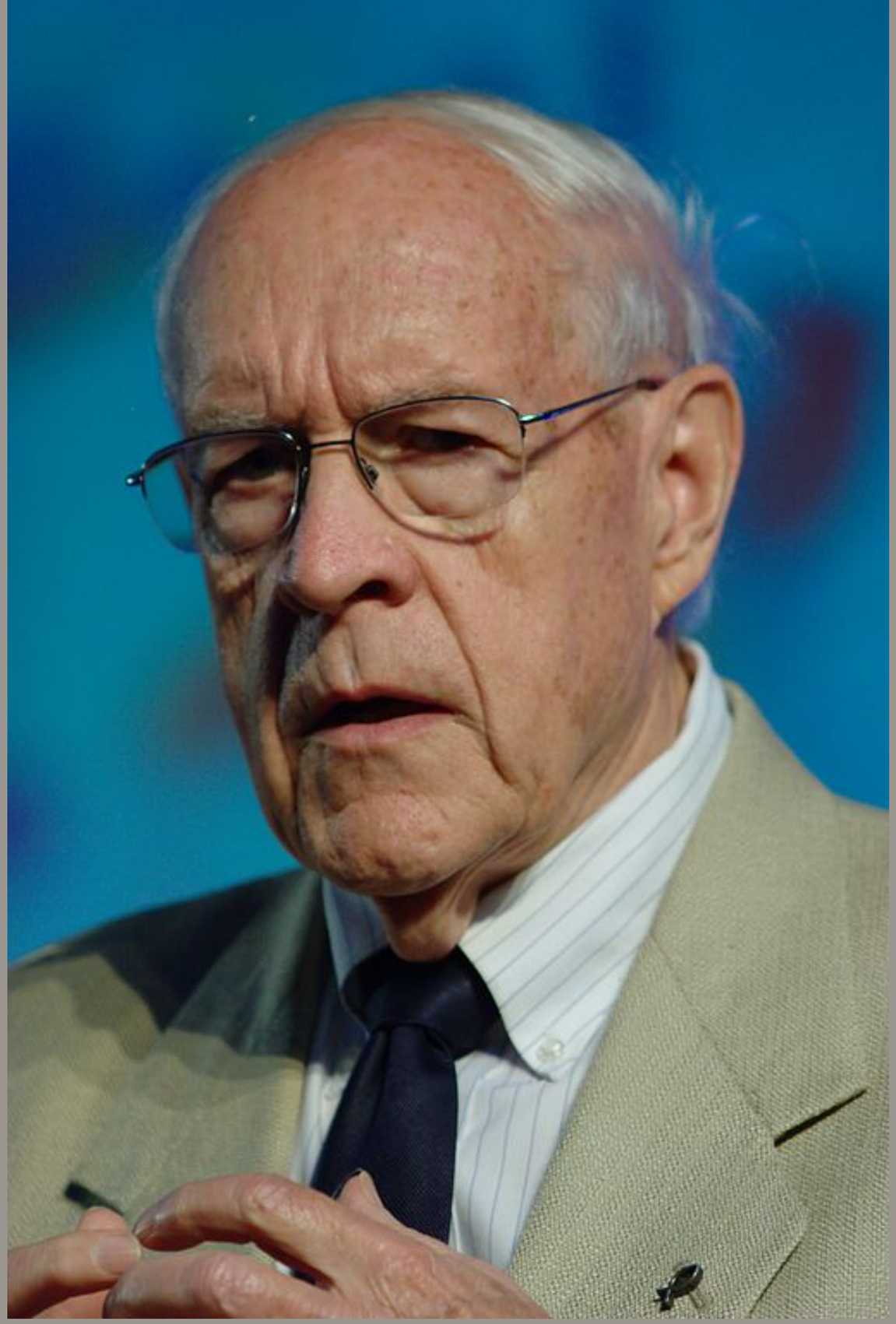
http://mobprogramming.org

https://www.industriallogic.com/blog/a-few-tips-for-mob-programming/

also:



@JLangr



https://commons.wikimedia.org/wiki/File:Frederick_Brooks_IMG_2279.jpg

Brooks' Law

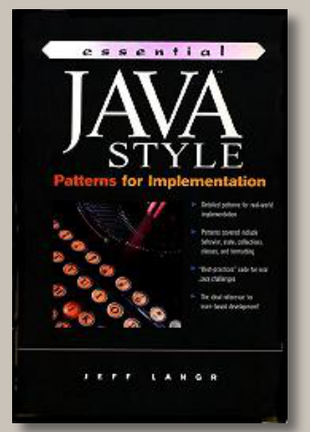
"Adding manpower to a late software project makes it later."

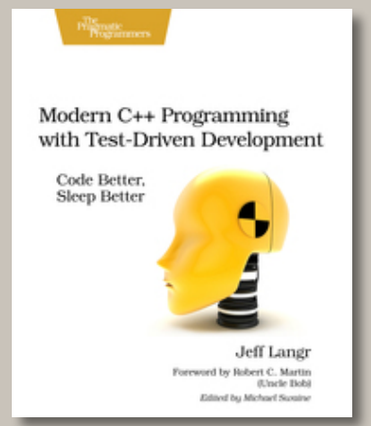
The Mythical Man-Month, 1975

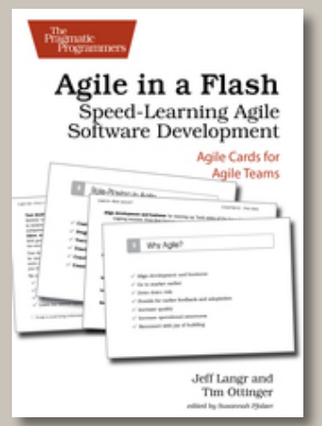
@JLangr

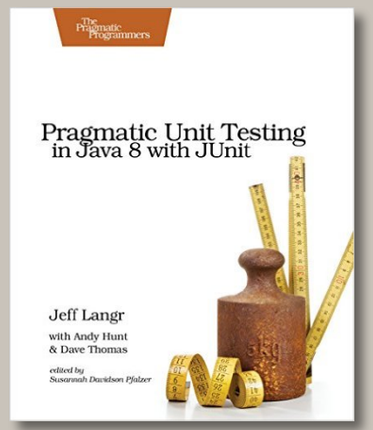
Thank you!

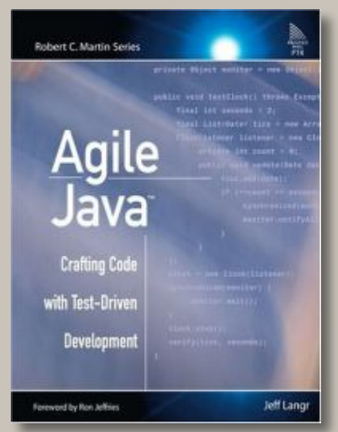
Langr
Software Solutions
SOFTWARE TRAINING/CONSULTING
TDD
AGILE

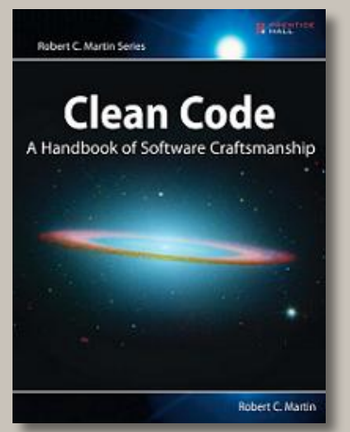

essential
JAVA STYLE
Patterns for Implementation
JEFF LANGR


Modern C++ Programming with Test-Driven Development
Code Better, Sleep Better
Jeff Langr


Agile in a Flash
Speed-Learning Agile Software Development
Agile Cards for Agile Teams
Jeff Langr and Tim Ottinger


Pragmatic Unit Testing in Java 8 with JUnit
Jeff Langr
with Andy Hunt & Dave Thomas


Robert C. Martin Series
Agile Java
Crafting Code with Test-Driven Development
Jeff Langr


Robert C. Martin Series
Clean Code
A Handbook of Software Craftsmanship
Robert C. Martin