

Automation Addiction



Hello!



We are
Huib Schoots & Paul Holland

@huibschoots | @PaulHolland_TWN

Disclaimer

The opinions expressed in this presentation are strictly those of the presenters and do not necessarily reflect the opinions of our employers *De Agile Testing* and *Plato Testing*.

What is automation anyway?

“

**Adding more unreliable
code to the stack of
code you already have**

--Unknown

What does automation mean?

- Activity: Data collection, comparing, mimic input, recording, data generation, generating load, ...
- Level in code: Unit, integration, system, ...
- Workflow: Task, end-to-end, user experience, ...
- Architecture based: Feature, interface (UI / API), internal business logic, database, ...
- Agile structure/terminology: TDD, BDD, Acceptance Tests, Epic Tests, Product Tests, ...



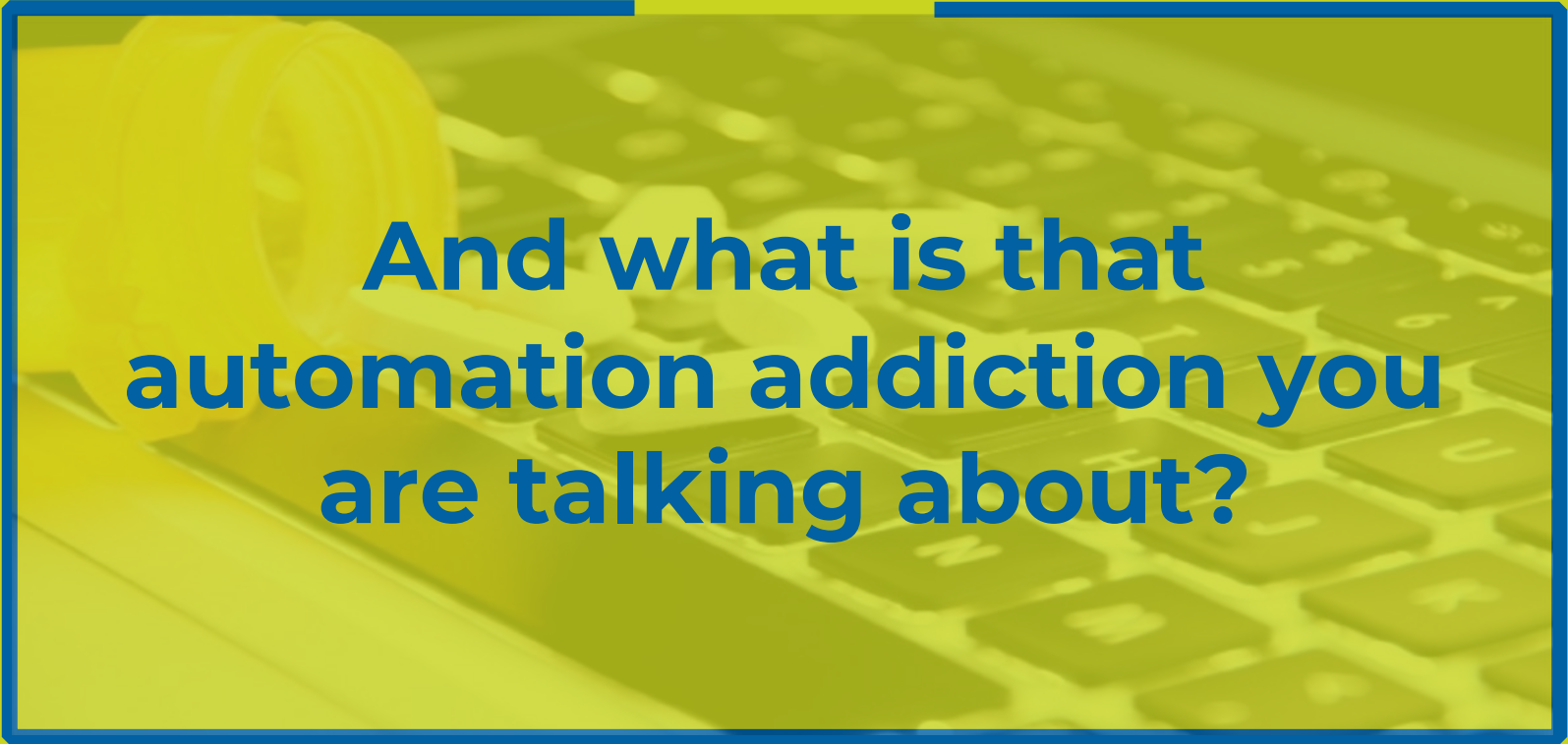
**So many types, styles, ways
of automation!!**

**We need to be specific
to have a good discussion..**

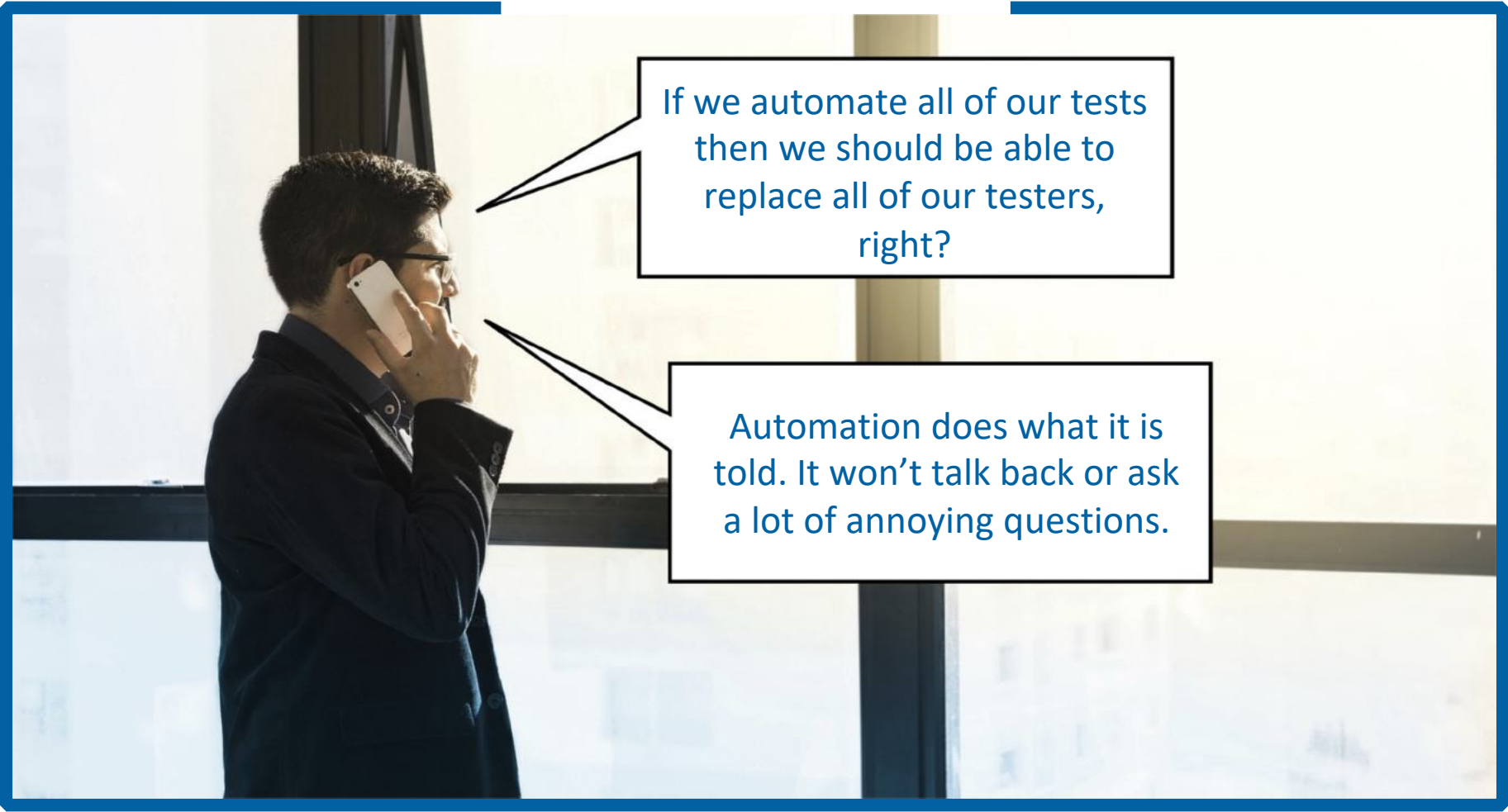


**Although we will do this
keynote without being specific**

**So, by our own logic, it won't
be a good discussion ... 🤔**




**And what is that
automation addiction you
are talking about?**



If we automate all of our tests
then we should be able to
replace all of our testers,
right?

Automation does what it is
told. It won't talk back or ask
a lot of annoying questions.




I wouldn't have to listen to
testers complain about the
poor quality of our products
any more.

Think of all the money we'll
save.

A yellow-tinted photograph showing a white pill bottle lying on its side on a laptop keyboard. Several white, oval-shaped pills have spilled out from the bottle. The text "18 months later ..." is overlaid in the center of the image in a bold, blue font.

18 months later ...



How did we end up on
the front page of the
New York Times with our last
release? We lost
\$2 billion in our market
valuation because of
that bug.

Symptoms of Automation Addiction

- ❑ Trying to automate too much (or everything)
- ❑ Trying to maintain ineffective automation
- ❑ Trying to replace testers (or testing)
- ❑ Implementation problems because some senior person likes the “tool or system”
- ❑ Sunk cost - refusal to get rid of something that is too costly or ineffective
- ❑ Testers who turn into SDET and only focus on automation

“

**Fixation on “test automation”
(and now on “AI”) is causing us to
lose connection with the human,
social purposes of software
development and testing.**

--Participant AiT training

Fairytale Marketing

mind.k

7 Ben

Hello Paul,

Hope you're doing well

By way of introducing our proprietary solution, we are introducing manual testing & automation.

This latest feature of artificial intelligence in QA Automator will get the application cost for any business to absolute zero, No more waiting for do all of this in the blink of an eye with a single click.

I have created a demo video, please have a look and its challenge to all my viewers provide me any application and witness the magic.

<https://lnkd.in/duqkQ8g>

Recently we helped [redacted] with your esteemed organization. Other [redacted]

Are you currently looking at ways to significantly reduce manual test efforts and development lifecycle?

Kindly suggest your available slots when we can demonstrate TestingWhiz and its powerful advantages that can add value to your esteemed organization.

Pravesh Jha • 1st
Test Architect || Creator of QA Automator
1w • Edited •

How about artificial intelligence in Automation testing !
An year back if anybody would have said that automation scripts will get automatically created with the single click and without application, people would have called him Day dreamer but today I am introducing this game changer in QA Automator.

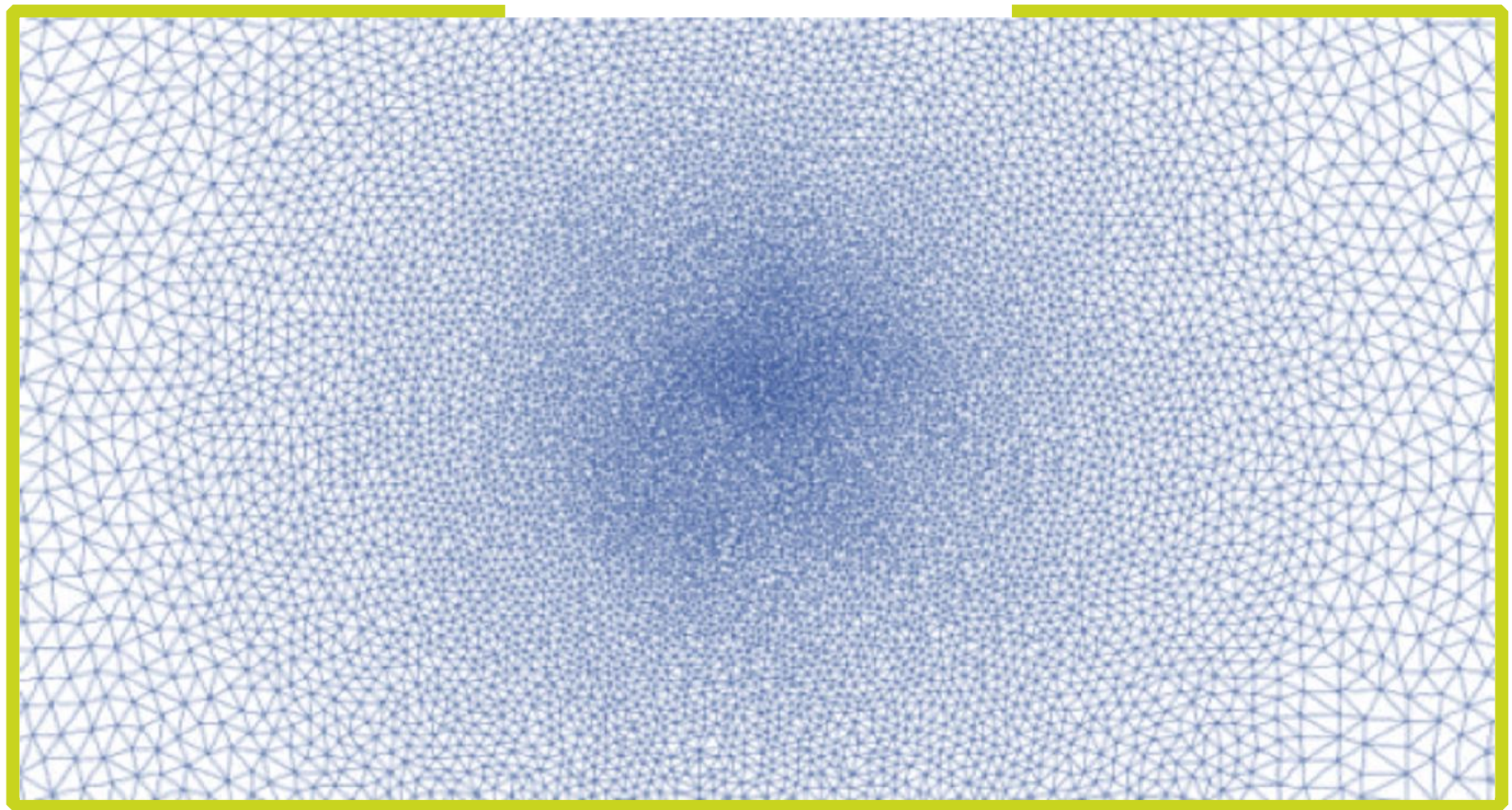
duct at
Tricentis
their testing
areas and
to cut
duce
0%.

ingWhiz
product at
er one
success-stories
/reducing-

id software
our esteemed



Why not automate
everything?



@huibschots | @PaulHolland_TWN



Unknown unknowns

- Customers/Product Owners don't know or can't imagine what they want...
- Development team can't image what customers will actually do
- Research: new insights & evolutionary design
- Dealing with complexity, confusion, change, new insights and half answers

We have to learn and deal with risks!



Story time!

This is my story!



Paul's Story

A large American Insurance Company:

- They had an EXCELLENT automation framework
- They had over 2500 scripts
- Testing all written requirements
- About 40% of scripts failed each execution
- 13 outsourced testers overseas in a different timezone took 2 weeks to execute them and investigate failures
- **On average this team found 6 defects a month**
- **I pair tested the product for 20 min and found 5 bugs**



And this is my story!

Huib's Story

A large Dutch Insurance Company:

- Huge project (100+ people)
- Good quality, happy customer
- 25 testers, 3 full time automation engineers
- Approx. 24K checks
- 1 major bug: 700K people got a letter with their monthly deposit calculated wrong
- **Testers relied too much on the automation and missed an important (and pretty obvious) bug**



What Automation Can and Can't Do

What can automation do for you (if done well)?

- Allow for CI/CD pipelines - **faster delivery!!**
- Do a good sanity check of the system
- Increase frequency of sanity/regression testing
- Enable Testers to look for deeper bugs
- Give you faster feedback to changes
- Jobs that are repeated often or are error-prone

What can automation do for you (if done well)?

- ❑ Reduce the number of Exploratory Testers needed - but you will likely need as many Automation Testers - so no cost savings (probably more expensive)
- ❑ Data collection, data input, and calculations
- ❑ System monitoring
- ❑ **Decrease** (certain) risks

What can automation do for you
(if done poorly)? (which is most
of the time)

- ❑ Cost increases
- ❑ Execution time increases - investigation of failures
- ❑ Product coverage decreases
- ❑ Demoralizes employees
- ❑ Trust in automation declines
- ❑ **Increases** (a lot of) risks

What automation will NOT do for you (even if done well)

- ❑ Replace the need for human testers. Although you could possibly reduce the number
- ❑ Decrease your costs - Tester reduction costs are replaced with SDET costs and H/W & S/W
- ❑ Increase your product coverage
- ❑ Save you time - writing, debugging, and maintaining the automation takes time plus you should still actually test for bugs



Problems with Automation

Automating too much

- ❑ Not thinking about what automation is helpful
- ❑ Thinking that more automation will find more existing bugs in the system on an ongoing basis
- ❑ Ignoring the future maintenance costs
- ❑ Automation Silos - Automating each story instead of coordinating across stories/epics/products

Automating the wrong things

- ❑ Automating something that can be done very quickly by people (UI correctness, for example)
- ❑ Automating unimportant tests because you can
- ❑ If a test fails and no one fixes the issue then it was not important enough to automate
- ❑ Automating tests at the UI level that should be automated at the API (or lower) level
- ❑ Lacking a good strategy to identify important tests and what should be automated

Sunk Cost Fallacy

- ❑ Not willing to discard ineffective, flaky or obsolete scripts
- ❑ Not stepping back and evaluating the CURRENT value of the old automation
- ❑ Not willing to start over with a better plan

Other problems with Automation

- ❑ Written by unskilled or unmotivated people
- ❑ No code reviews or static analysis
- ❑ Scripts created independently by different teams instead of a unified approach
- ❑ Using a tool because *"it is the tool you know"*, *"you were told to"* or *"we paid for the license already"* and not because it is the right tool to use
- ❑ Automation not part of a daily routine or pipeline (when applicable)



Okay, I believe you

**But how do I convince my
manager/team/organisation
about this??**

How to talk about automation?

We (Huib and Paul) are experienced testers with consulting experience and over 50 years of testing

**HOLY CRAP! YOU GUYS
ARE OLD!**

- We often fail at convincing companies, management or teams that their approach should be improved/changed
- We have had some successes - we will cover those now

If you two can't do it, then why should I try?

- To be heard, we need more voices speaking out
- Unlikely to break an addiction in a single conversation
- You need willingness of others to change (repetition)
- Don't try to convince the management, but your team
- Just do it: start small and improve

Here are some ingredients that can help you...

Ingredient 1: Why keeping Exploratory Testers is important

- Automation repeats the same checks
- Automation (typically) does not test deeply
- Automation will not find bugs that it already missed
- **Think of an Easter egg search**

Ingredient 2: The Importance of Test Strategy

- Should help create scripts that test the entire system across multiple teams/products
 - Reduces the execution time
 - Reduces maintenance
 - Reduces redundancy
- Choosing the right tool or framework and not just using the tool you were told to (or the tool you already know)
- **A good strategy should help overcome sunk cost fallacy**

Ingredient 3: Don't try to automate everything

- Execution time will increase over time
 - Effectiveness of time to execute decreases
- **Scripts provide no insight (need people for insights)**
- **Maintenance costs are often overlooked** and will climb
- Do not automate checks that will not get fixed if they fail
- **Lose the “customer focus” overview of testing**
- Remind people of the insurance company stories

Takeaways

- ❑ Robots cannot replace human thinking, creativity or reasoning (think: find new bugs vs sanity check)
- ❑ Try to notice “bad automation” situations early to change direction before sunk cost fallacy occurs
- ❑ Remember what automation can and can't do
- ❑ Do not try to automate everything
- ❑ Have a comprehensive strategy supported by the whole team (increasing testability)

Automate responsibly!

Questions?
Remarks?
Feedback?
Stories?

More info

- ❑ Rapid Software Testing - <https://rapid-software-testing.com/>
- ❑ Great Resources - <https://www.huibschoots.nl/links>
- ❑ Paul Holland – Skilled based testing video - <https://youtu.be/piNigC1cRV4>

Thank you

These wonderful people helped review this presentation:

Vincent Wijnen, Peter Schrier, Joost Voskuil

Special thanks to all the people who made and released these awesome resources for free:

- ▣ Presentation template by [SlidesCarnival](#)
- ▣ Photographs by [Pexels](#)

Contact



Huib Schoots & Paul Holland

@huibschoots

huib.schoots@gmail.com

www.huibschoots.nl

+31 6 24 64 10 33

@PaulHolland_TWN

paul@testingthoughts.com

www.testingthoughts.com

+ 1 407 666 0441

That's all folks!