* The purpose of this document is simply to collect all ideas I had during the Intland codeBeamer training in Stuttgart, 2019.05.28-29. Just write it all down as a „material”. Do not criticize, just get it out of your system. It’s not you, it’s your creative genius anyway.
* Look at an actual app on your phone. Reverse engineer what customer requirements it serves. How did it translate into system requirements?
* Create an application for restaurant orders.
  + Customer requirement: Get a list of restaurants.
* Create teams. Assign the “create your app” to the teams. It could also be something else.
* An example could be: develop an application for taxi orders called “Huber”. Customer requirement: I want to be able to order a taxi to the location where I am. I want to be able to order for a later date. I want to see where the car is and how long I will have to wait. I want to know who the driver is and what the car’s license plate number is.
  + System requirement: map integration with cars symbolizing the positing of vehicle.
  + Confirmation message in the app with a photo of the driver, type, color and reg. number of the car.
* Emphasize the importance of “simple examples”.
* CALL: The auditor is coming to our place next Friday. He wants to check the status of a release as of today. Let’s use codeBeamer baseline to accomplish this goal.
* codeBeamer came across as a likeable solution. I am quite surprise that it has been re-imagined for the retina look and feel, at it’s already pretty damn good.
* Training has to be people-focused. This was also mentioned by the smooth talking dude. One who has ears should hear the truth in words.
* My passions:
  + Thinking hats
  + Mindfulness
  + onlineLearning
  + Train the Trainer
  + Coaching
  + Breakfast with Gregory
  + Radio host / podcast style
  + English for non-native speakers
  + German conversationalist
  + Singing, Karaoke. Guitar. Forming a band. Hosting karaoke nights.
  + Presentation skills training. preZen-tation.
* Stand not sit during training.
* Look at people, not computers
* For every exercise:
  + Business case
  + Connect to big story
  + How do you use it?
  + Milestone of completion.
* Technical Shit:
  + I am an admin, a new user joins the team …
  + My boss gave me this diagram on desired workflow …
* You can say the same thing in different ways:
  + We are going to create a new role “external” OR
  + Imagine a new contractor joins your organization. You are the sysadmin responsible to set him up as tester in Scrum Team X.
* When it comes to training and Greg’s role in Intland I will be
  + Product Owner
  + Product Manager
  + Product
  + Sales
  + Marketing
* Clearly define the difference between:
  + Setup
  + Configuration
  + Customization
  + Integration
  + Administration
  + Transaction processing
* Tell the story of data.
* Demo – Contemplative Clicking – Slides
* Often say during the training: “Imagine…”, “Let’s say we are…”
* Do not assume people are already committed
* Names, introduction. Social. Let the training be a social event. Networking and collaboration.
* Presales is used to fast-pace demo.
* Use the second trainer in the room as a helper. Ask him to move his ass.
* Get a speaker’s stand and use it. Keep standing and project your voice. Connect with the audience via your eyes.
* Language preferences of the audience: they want German and get English training.
* Talk more about the training, less about the product.
* Always be explicit whether the audience should watch and listen or follow along.
* Do you guys have questions? Never. Ever.
* Training – students have to be instructed to register for codeBeamer cloud instance in advance.
* Disruptive training delivery.
* Intland codeBeamer training manifesto.
* Grow into a German-language trainer.
* Customer site visit for training preparation. Talk to the highest-ranking participant from each company before the training on their expectations. Who is the training for? What are their objectives? Discover that.
* Training and certification could hugely impact customer awareness of the product.
* Screenshots on slides??? WTF? The best would be to place links on the slides that take you to the next slide that represents the window that pops up in the application. You end up building and education version of the system instead of … clicking slowly in the real one.
* Learn the terminology of key industries. Automotive Quality Control. ALM. SCM (Source Code Management!)
* Take photos. Provide certificate of attendance. Talk about the next steps.
* Laci and Dani were so cool it almost hurts. Got 2 be a millennial for this!
* Gregory’s Training Manifesto
* Pre-class assignment.
* Demonstrate the difference between current training and planned training using requirements and tests. Instead of codeBeamer can do this – use the word imagine a lot more.
* Objectives, validation, training.
* Information from students?
* Teams should choose an application and present codeBeamer using that as a Case Study.
* “codeBeamer helps you organize the efforts of a team to achieve a common goal”
* Agile is not a criticism of waterfall, nor is Mac critical of Windows. Simply beg to differ.
* Games should take place after all main subjects.
* Survey potential students. What is it that you want to learn? Agile? What is advanced training?
* Too much time – use assignments!
* Release / Customer Requirement / System Requirement / Story Point
* Employ people’s visual memory.
* Picture a typical team in a project. Create that in advance in codeBeamer.
* The more experience one has the less likely they are to use easy-to-imagine examples.
* Close your eyes. Imagine an app on your phone. What can you do in that app? What purposes does it serve? These are the customer requirements.
* Open the app. Look at its screens. How do screens meet those requirements? What should screens have to meet those requirements?
* Look at audience not slides.
* Roles instead of groups.
* When showing reports talk about business value. What information can we get that we did not have before. And here is a velocity report. The project manager wants to see the changes in team performance in the last months.
* One in this business should be familiar with automotive control systems, processes and tools.
* Surely CB can do a whole lot of things and can be hugely customize. But what can it do out of the box? What purpose does it serve?
* Write down all of these notes in your cool MacBook Pro. Truth is: it’s your material for work.
* Practice English with mic. The American Yeti project.
* codeBeamer lends itself well to teamwork and games. Teams can share projects.
* Strategy: silence. You feel like sharing – share.
* Remember your own policy – S4A – Setup for Action.