Engineering for Software

How to Amplify Creativity

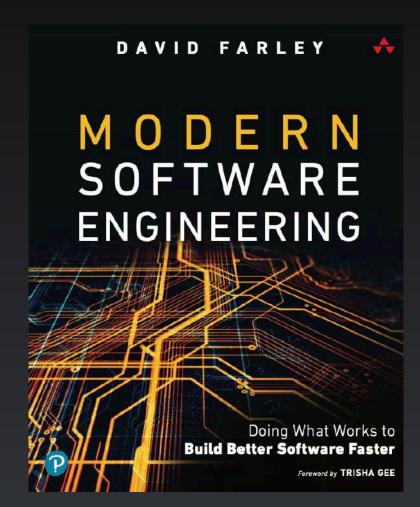
Dave Farley

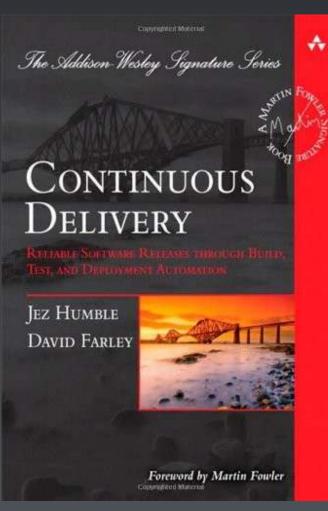


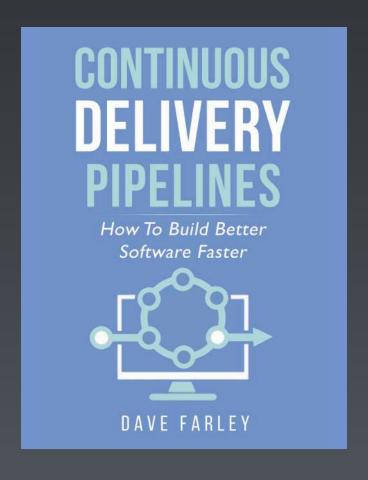












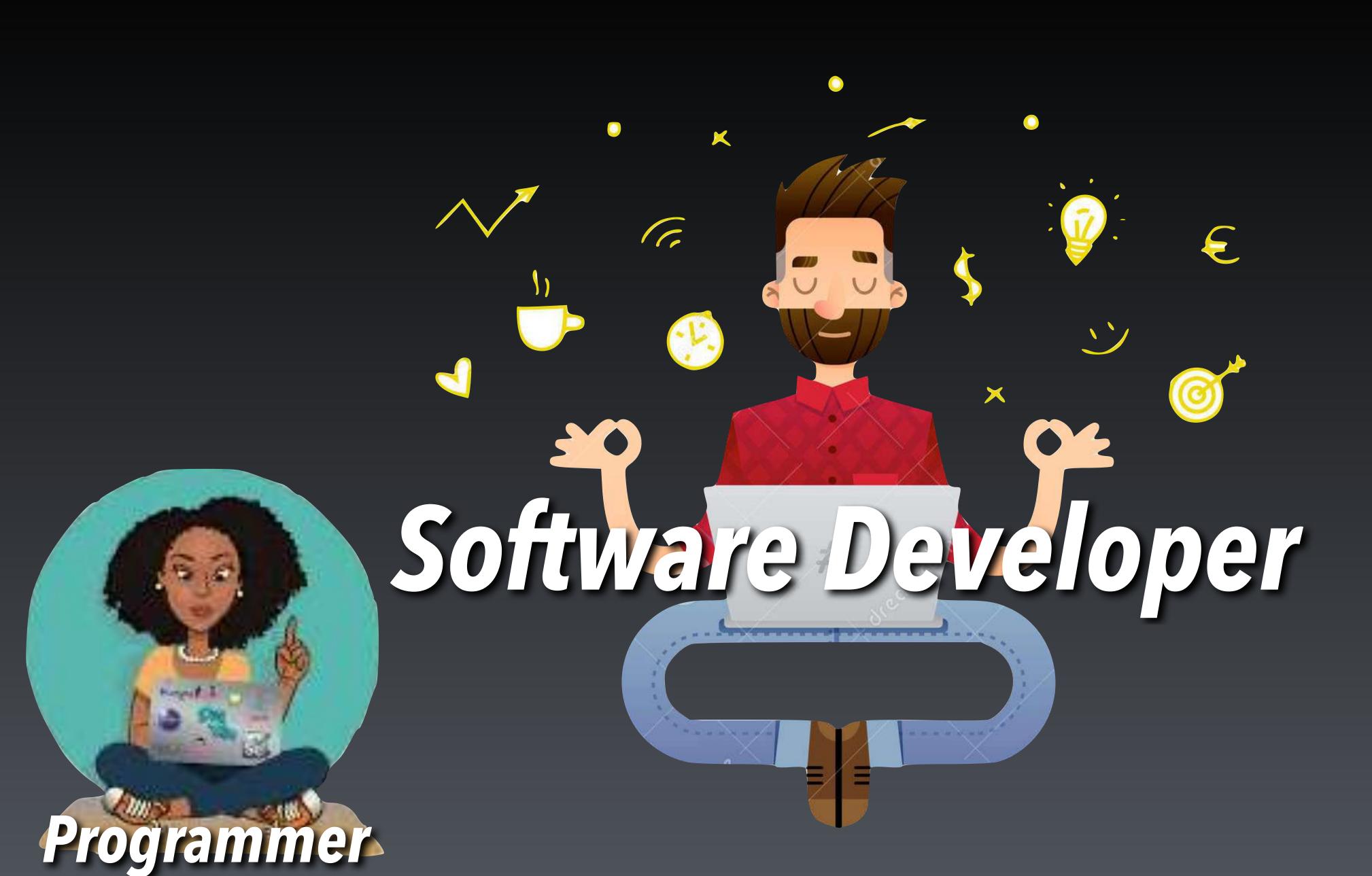


How Do You Think of Your Work?

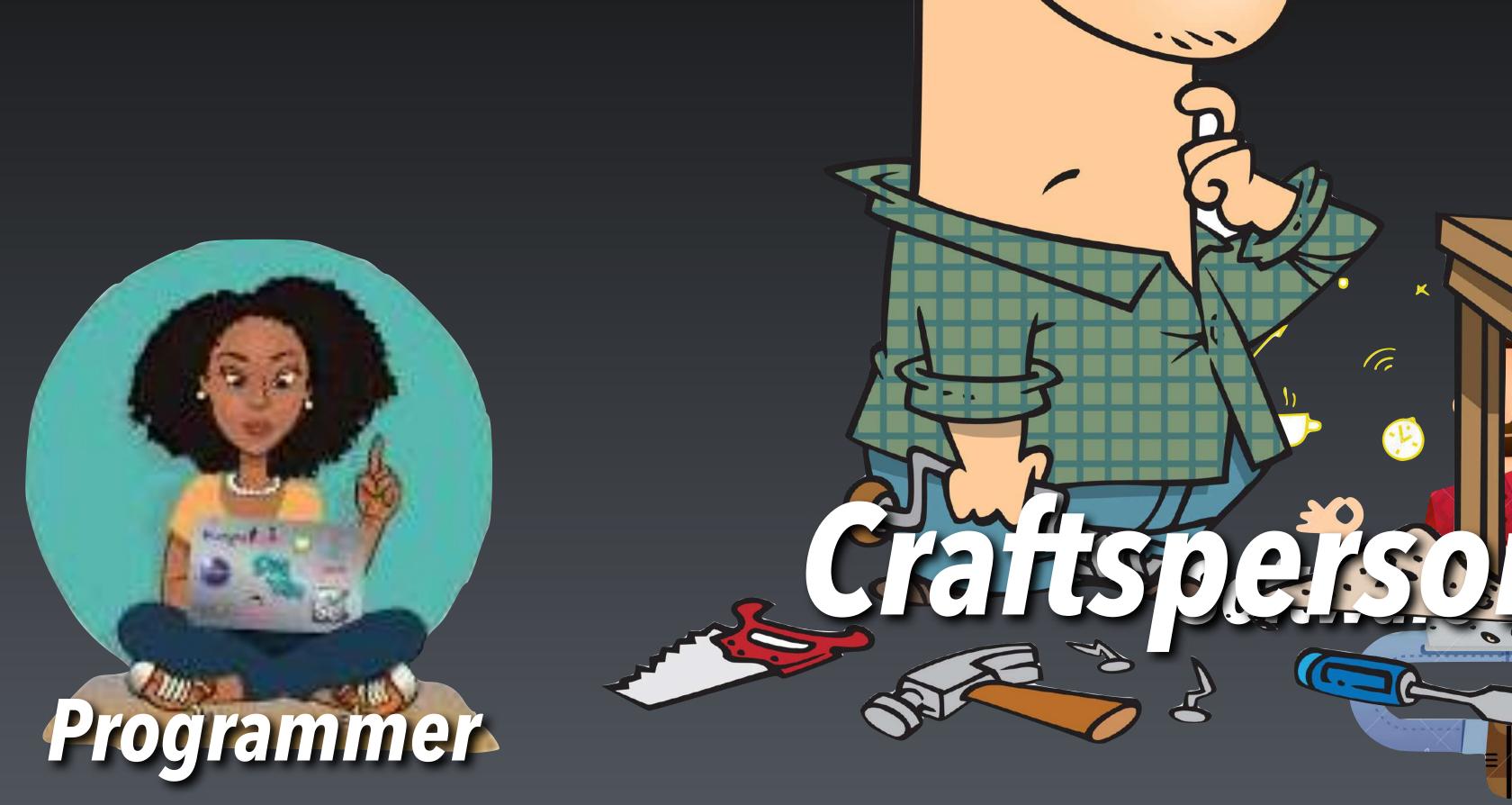












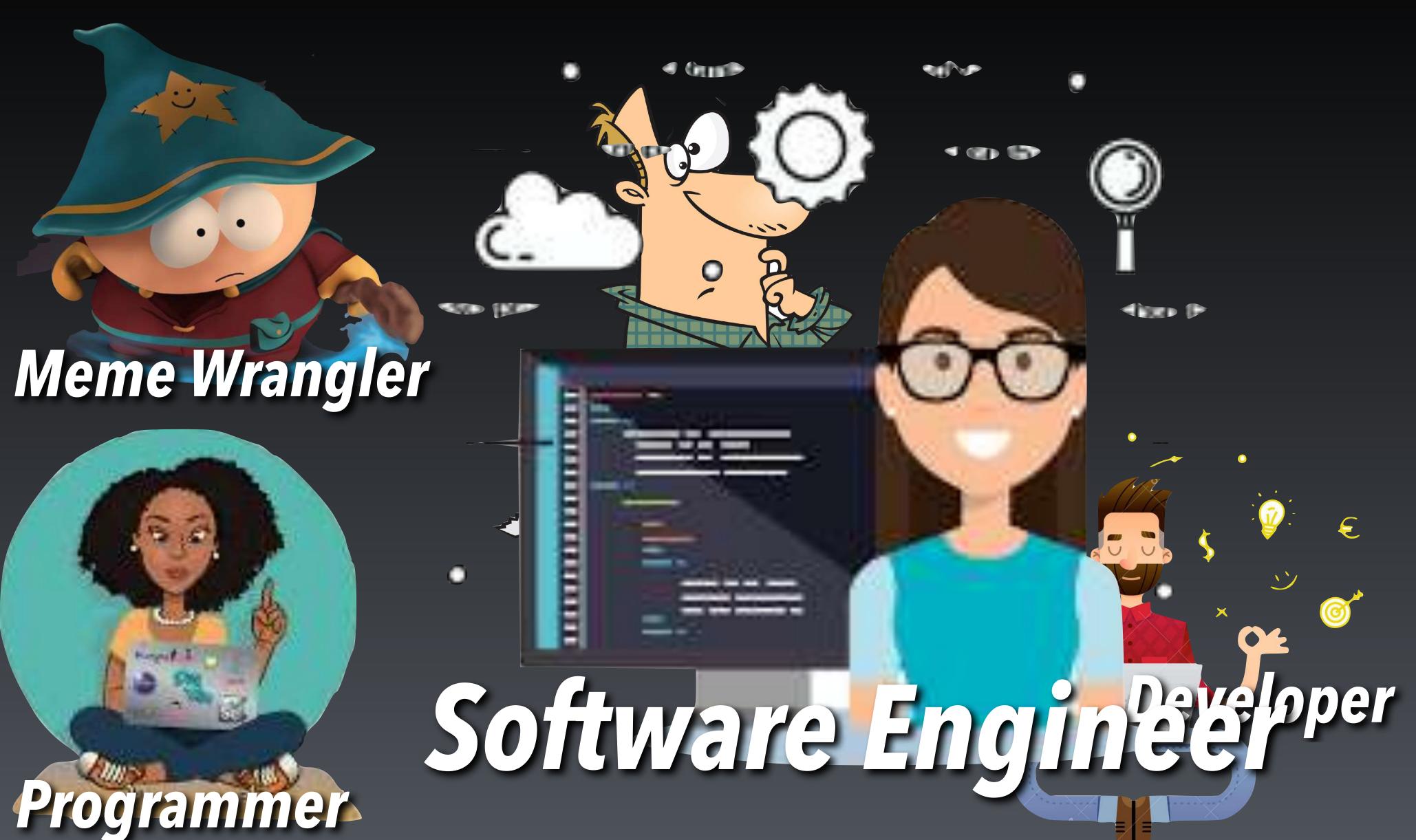








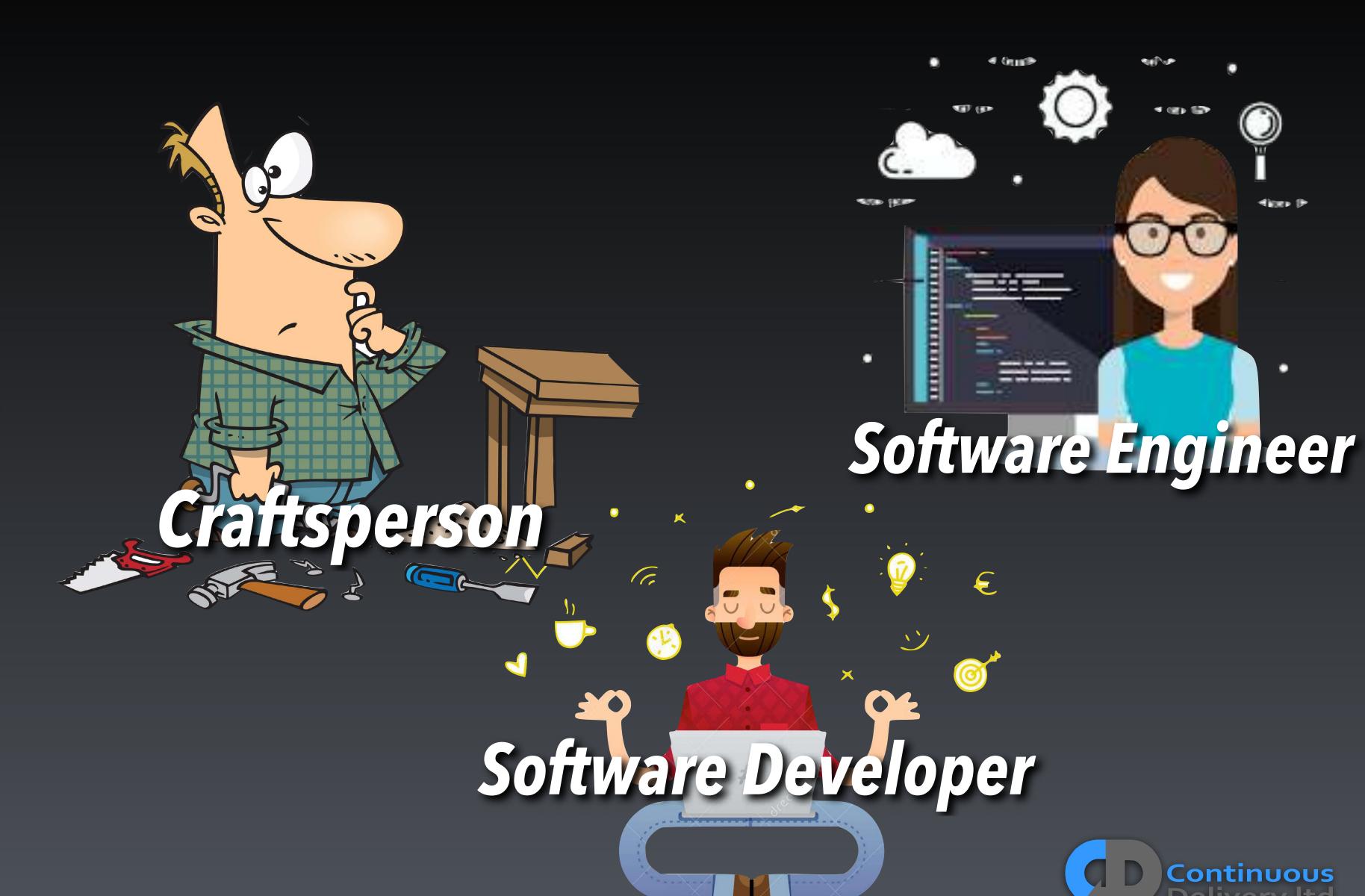












Continuous Delivery Itd

Who Cares?



Who Cares?

Some of These Things are More Than Only Labels





· What is Engineering



· What is Engineering

· Common Foundational Principles



- · What is Engineering
- · Common Foundational Principles
- 10 Guiding Principles for Software Engineering



- · What is Engineering
- · Common Foundational Principles
- · 10 Guiding Principles for Software Engineering
- · Applying the Guidelines to Code



What is Engineering





• Engineering in Other Disciplines is: "The Stuff That Works"



• Engineering in Other Disciplines is: "The Stuff That Works"

• Software is Difficult!

So What "Works" for Us?



All Engineering is not the same!

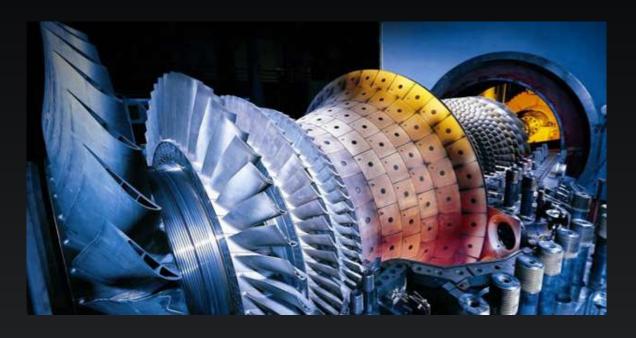






All Engineering is not the same!

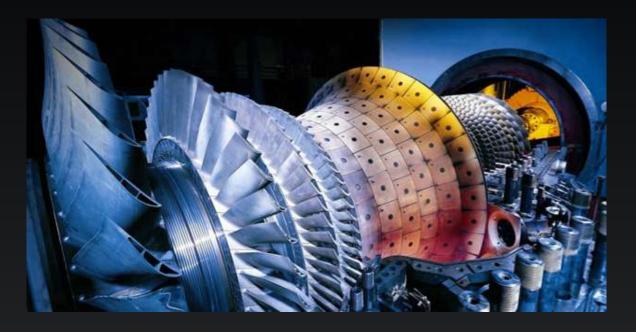






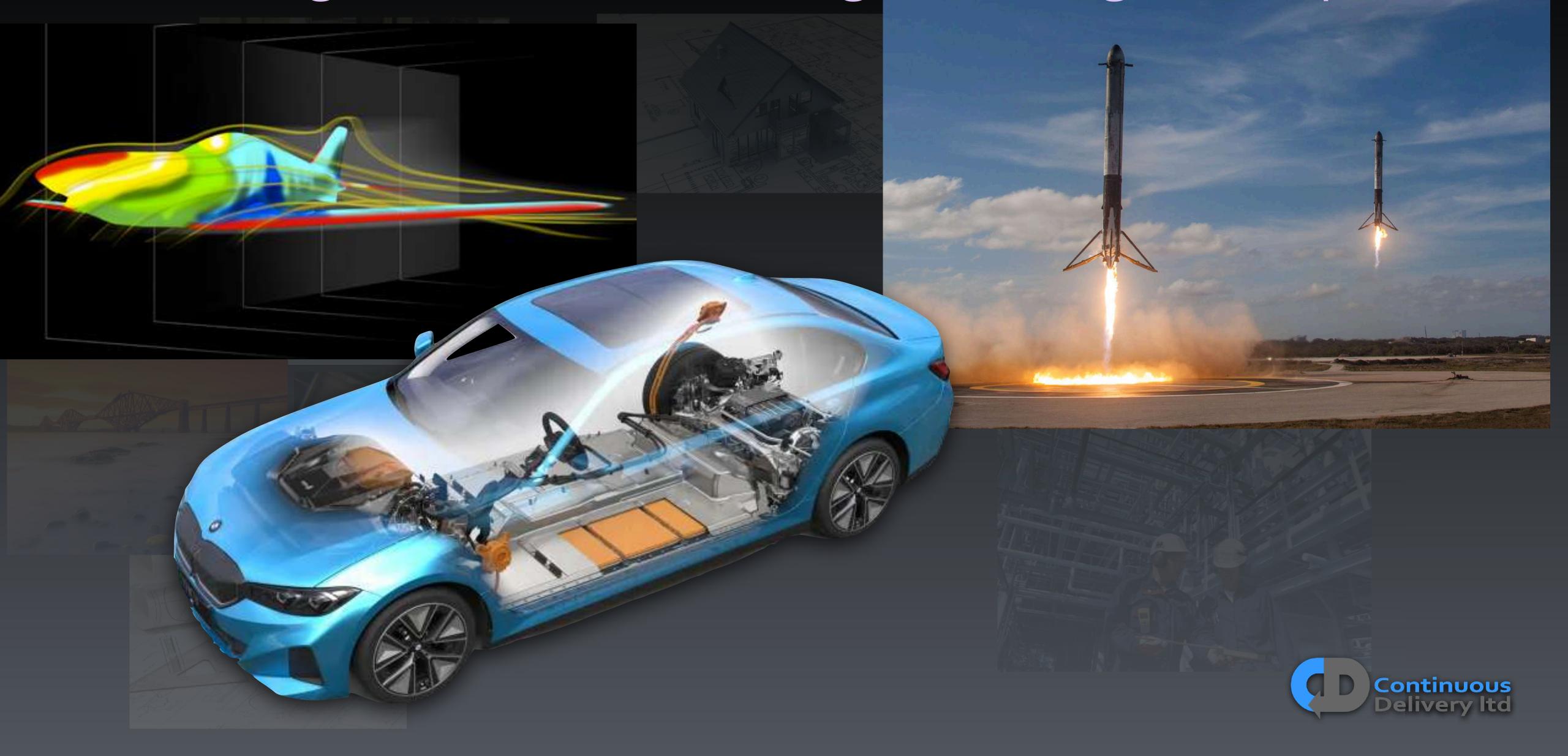


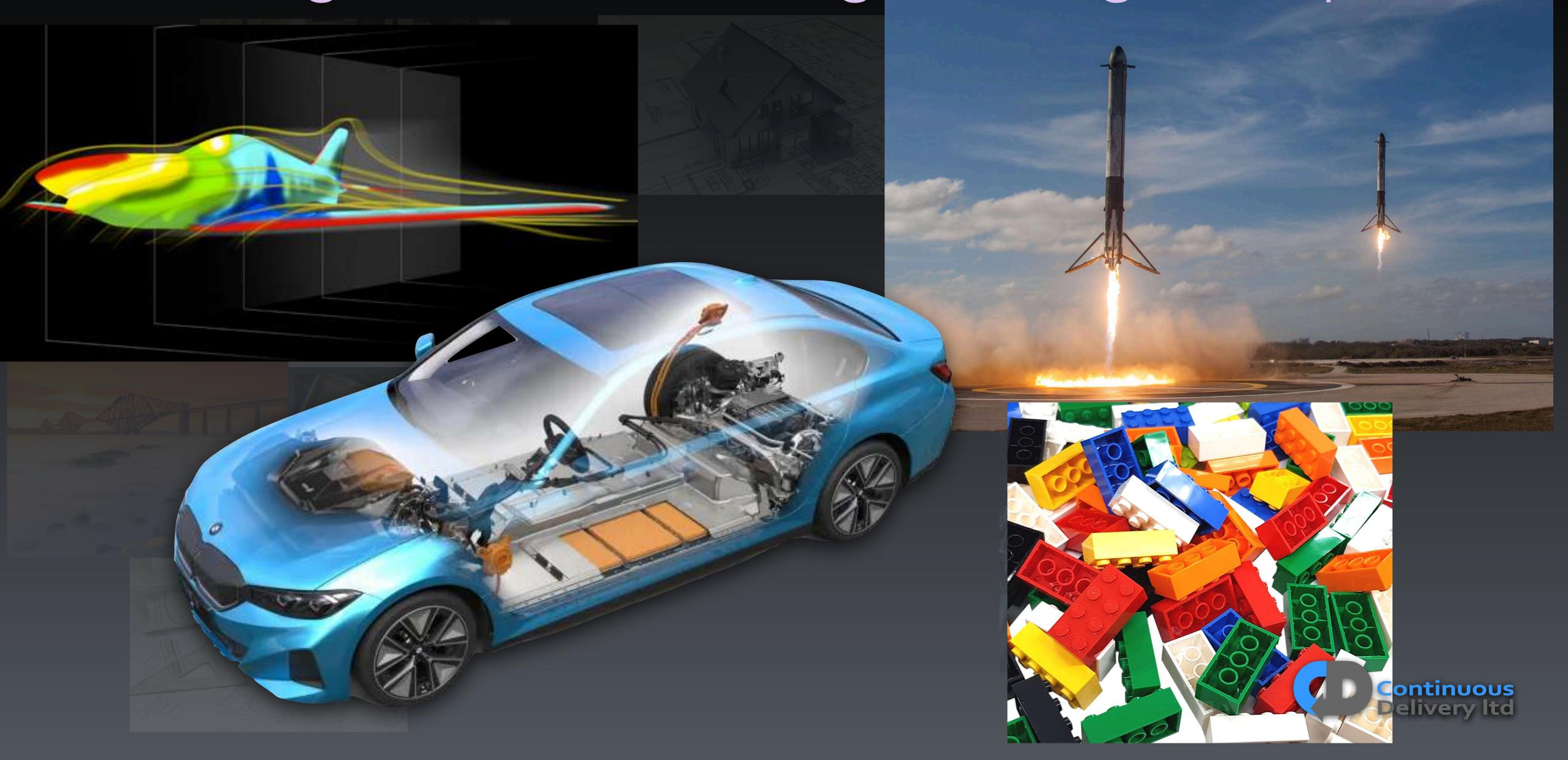














What is 'Engineering'?



What is 'Engineering'?

Engineering is the application of an empirical, scientific approach to finding efficient solutions to practical problems.

(Dave Farley - Just Now!)







is How We Solve the Hard Problems!





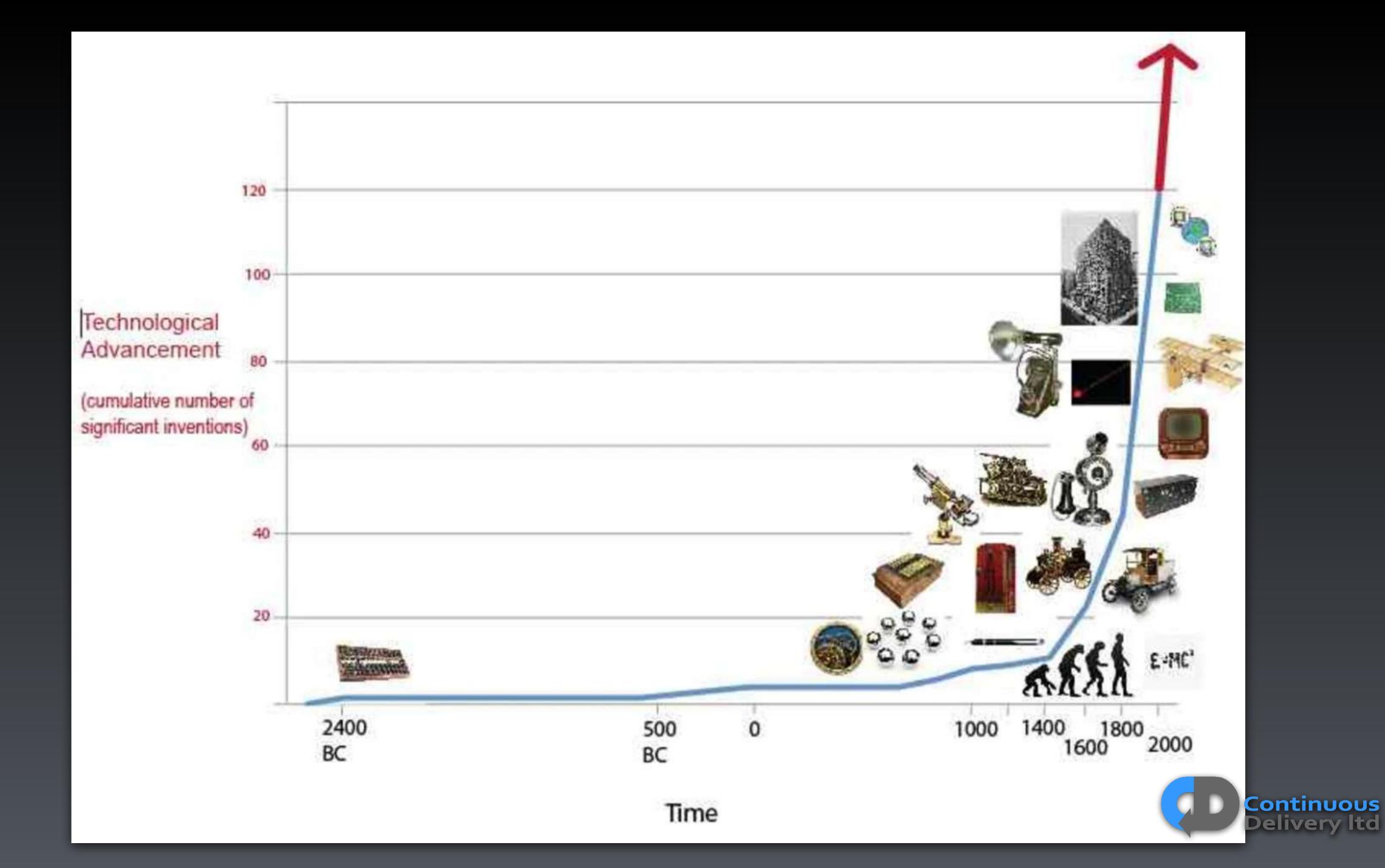


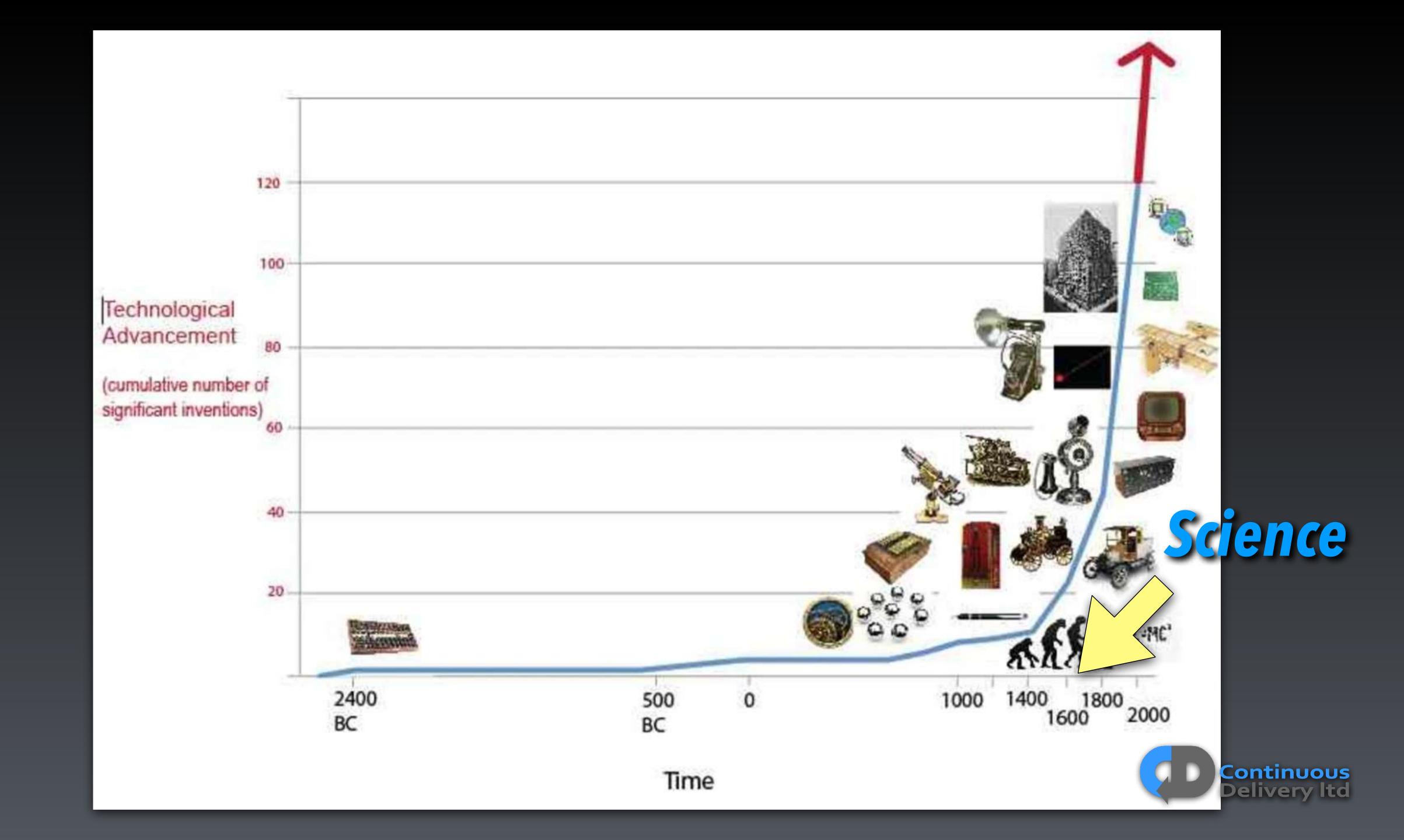
Engineering

The Foundation on Which Our High-Tech Culture is Built!

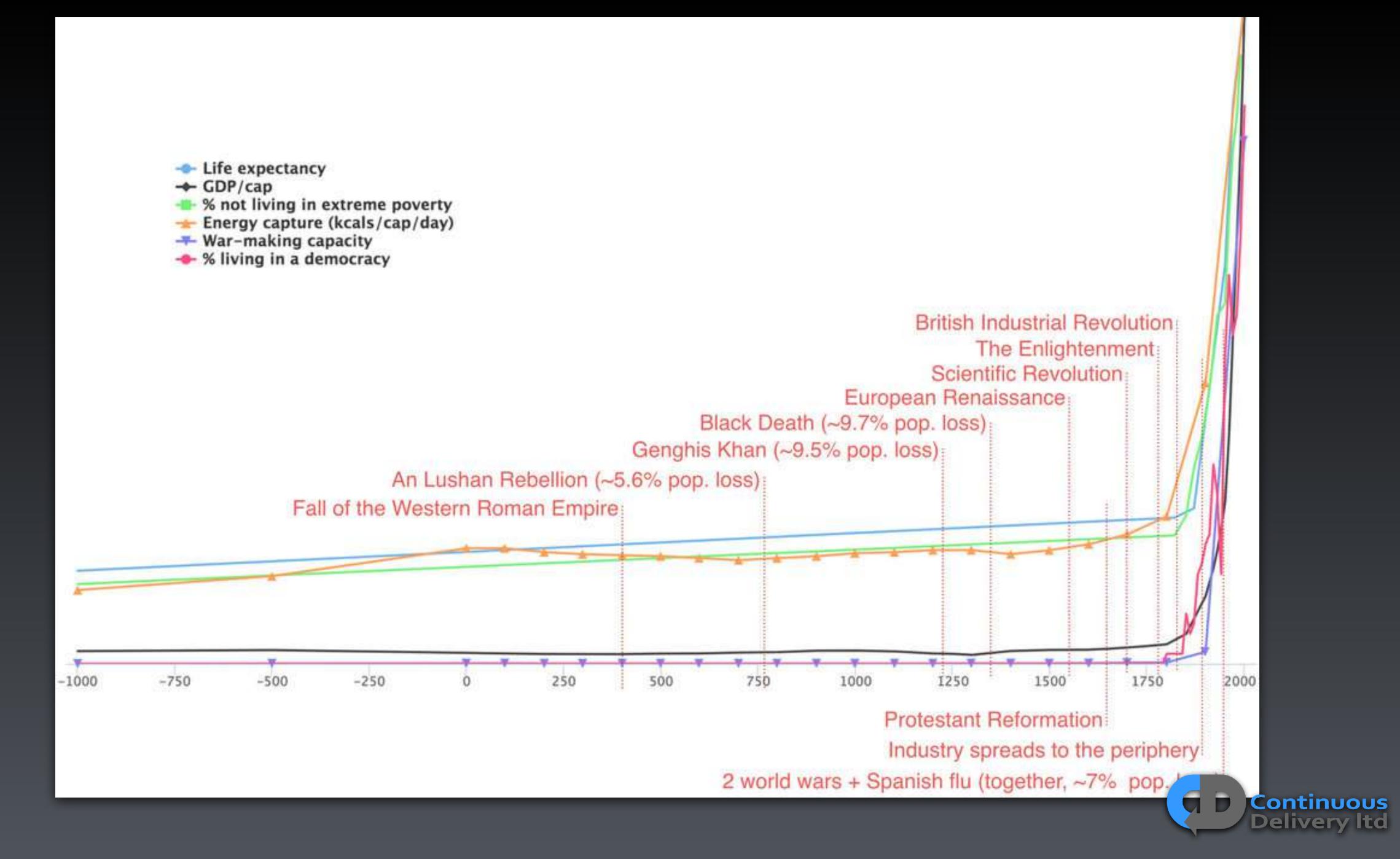


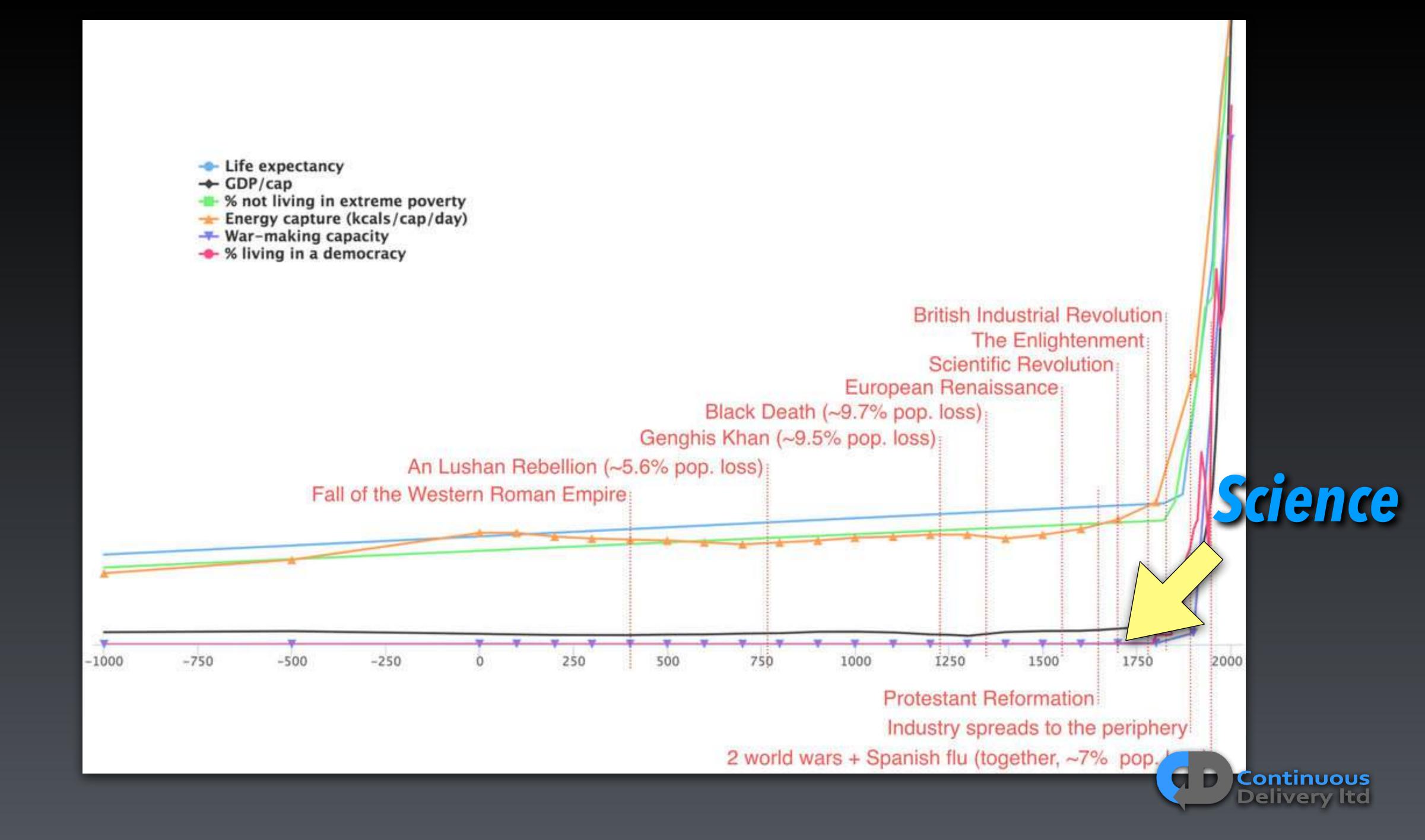




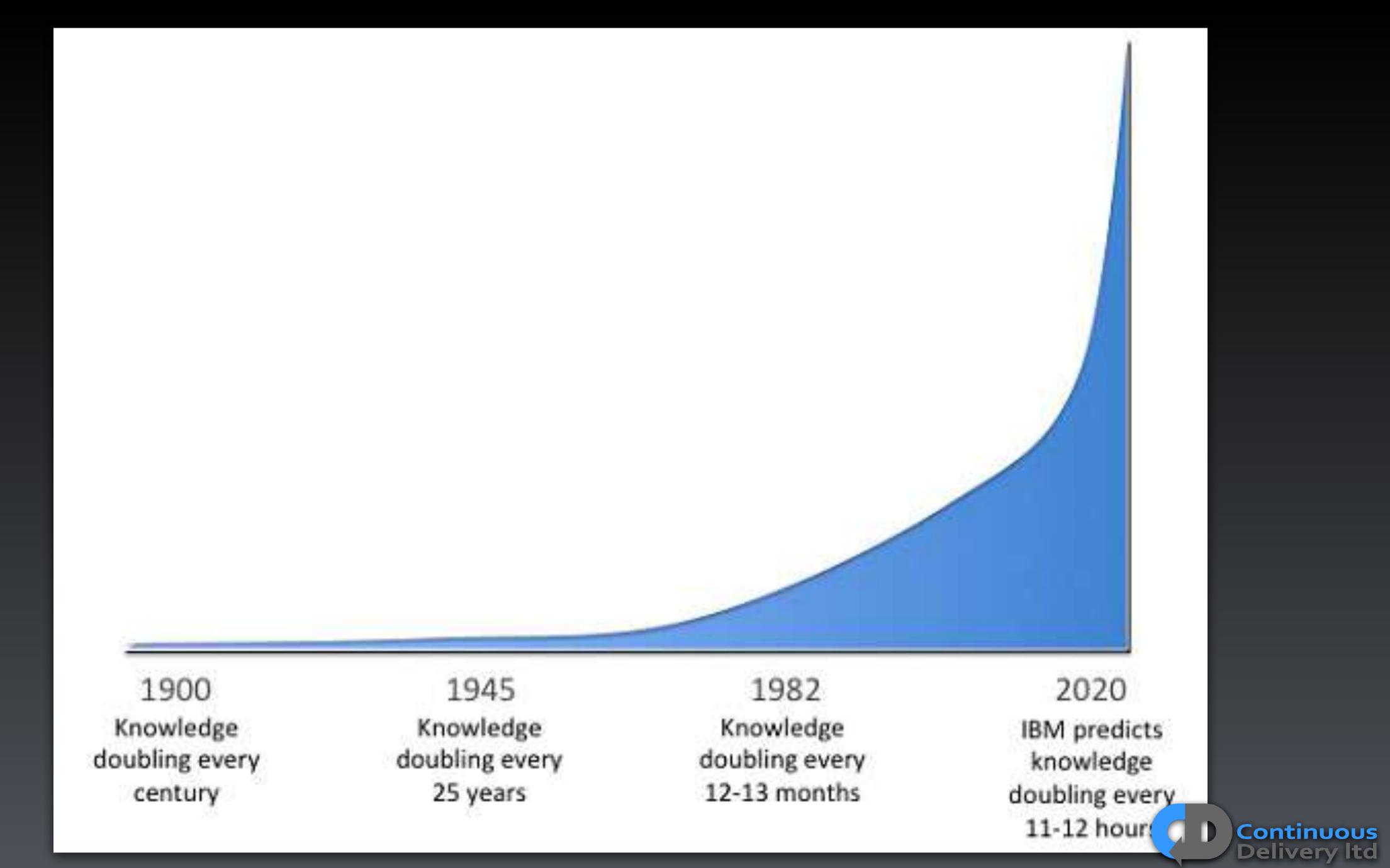












Science

1900

Knowledge doubling every century

1945

Knowledge doubling every 25 years

1982

Knowledge doubling every 12-13 months

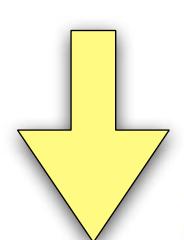
2020

IBM predicts knowledge doubling every 11-12 hour



Computers & Software

Science



1900

Knowledge doubling every century

1945

Knowledge doubling every 25 years

1982

Knowledge doubling every 12-13 months

2020

IBM predicts knowledge doubling every

11-12 hour





What is Engineering Really For?







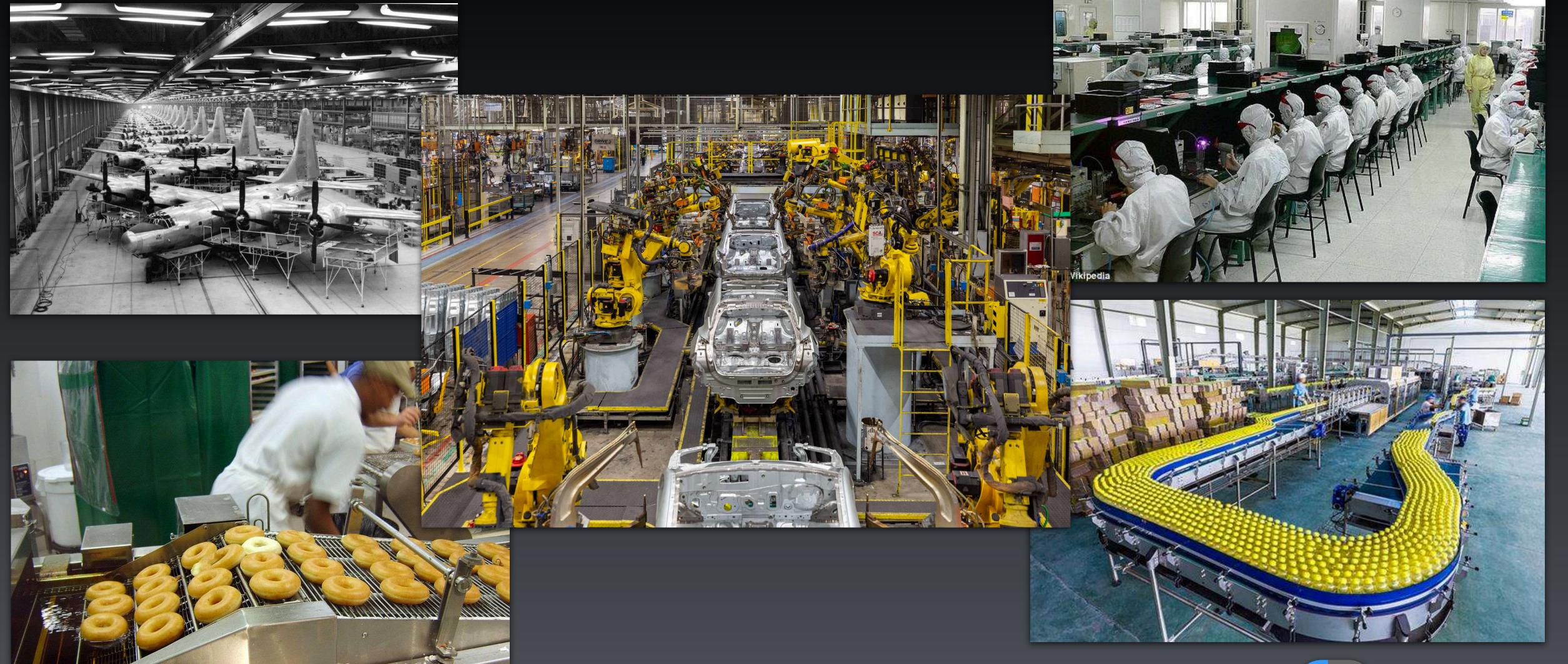




"The Things We Can't Afford to Get Wrong"



"The Things We Can't Afford to Get Wrong"





"THE TIMES WAS CALLED WILLIAM TO THE WILLIAM TO THE WAS THE WAS THE WAY TO TH



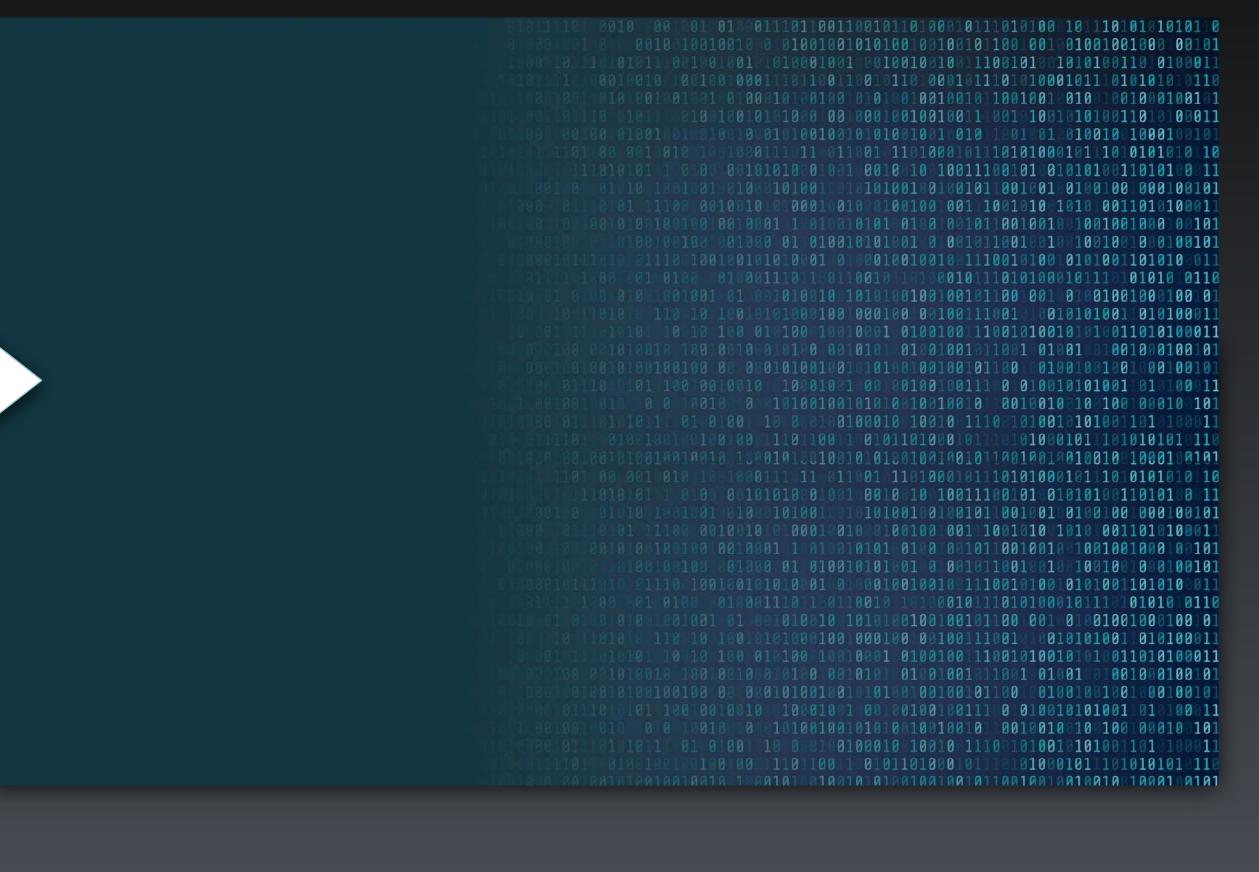




```
.
                                                      acceptance-testing - BookShoppingDsl.java [cd-acceptance-testing_test]
                                                                                                   🔨 🌃 Add Book To Basket Test 🔻 🕨 🇯 👣 🔲 Git: 🌠 🗸 🕥 🖒 📭 🔚 🔾
 acceptance-testing src test java com cd acceptance dsl BookShoppingDsl mcheckOut
                     🕀 🛨 💠 — 🔞 BookDepositoryBookShopDriver.java × 🔞 BookShopDriver.java × 🔞 BookShopDriver.java × 🔞 MyLocalBookStoreBookShopDriver.java ×
                                         package com.cd.acceptance.dsl;
   ▶ 📜 .gradle
   ▶ ■ .idea
                                          import com.cd.acceptance.dsl.drivers.BookShopDrivers;
    ▶ ■ gradle
                                          public class BookShoppingDsl
    ▼ Isrc
      🕨 🏣 main
                                             private final BookShopDrivers driver;
      ▼ lest tes
        ▼ <u>■ acceptance</u>
                                              public BookShoppingDsl(BookShopDrivers drivers)
          ▼ 🗀 com
             ▼ 🖿 cd
                                                  this.driver = drivers;
               ▼ 🖿 acceptance
                 ▼ 🖿 examples
                                              public void searchForBook(String... args)
                      ExampleMyLo
                                                  Params params = new Params(args);
                 ▶ 🖿 utils
                                                  String title = params.Optional( name: "title", defaultValue: "Continuous Delivery");
          ▼ 🛅 com
             ▼ 🗀 cd
                                                  driver.searchForBook(title);
              acceptance
                 ▼ Da dsl
                   ▶ 🖿 drivers 22
                                              public void selectBook(String... args)
                                                  Params params = new Params(args);
                                                  String author = params.Optional( name: "author", defaultValue: "David Farley");
                      Channel
                      Channels
                      CucumberDsl
                                                  driver.selectBook(author);
                      C Dsl
                      TestAnnotatio
                                              public void addSelectedItemToShoppingBasket() { driver.addSelectedItemToShoppingBasket(); }
          ▶ ☐ feature
        resources
                                              public void assertItemListedInShoppingBasket(String... args)
      w build.gradle
                                                 Params params = new Params(args);
      ■ gradlew
                                                 String item = params.Optional( name: "item", defaultValue: "Continuous Delivery");
       gradlew.bat
      🖾 run-selenium.sh
                                                  driver.assertListedInShoppingBasket(item);
      a settings.gradle
  ► Illi External Libraries
    Scratches and Consoles
                                              public void checkOut(String... args)
                                                  Params params = new Params(args);
                                                  String item = params.Optional( name: "item", defaultValue: "Continuous Delivery");
                                                  String price = params.Optional( name: "price", defaultValue: "£10.00");
                                                Card card = parseCard(params.Optional( name: "card", defaultValue: "1234 5678 9101 0001 12/23 007"));
                                                  driver.checkOut(item, price, card);
                                              public void assertItemPurchased(String... args)
                                                 Daname naname - man Daname(anne)
IntelfiJ IDEA 2020.2.1 available: // Update... (yesterday 19:58)
                                                                                                                                    48:82 LF UTF-8 4 spaces 🗜 master 🚡 👮
```

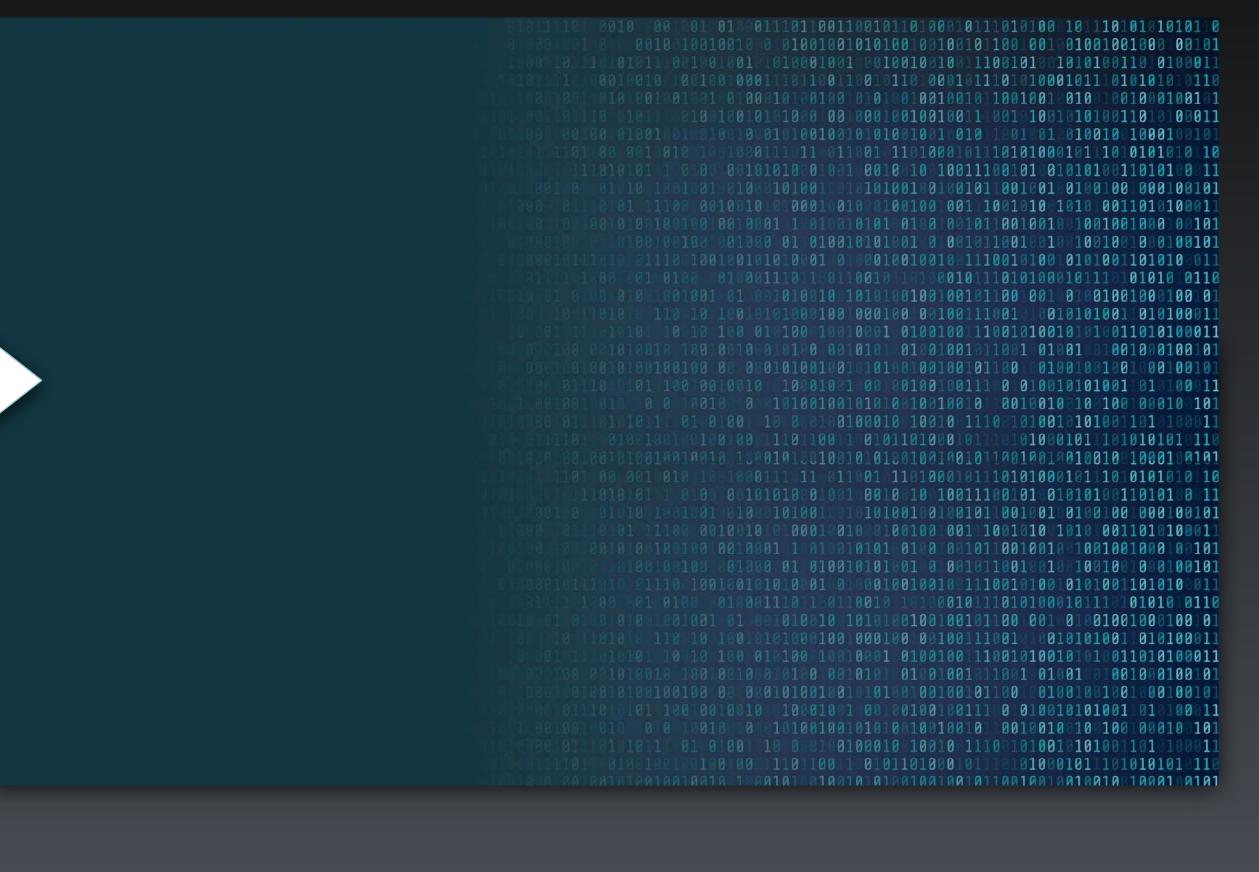


```
. .
                                                       acceptance-testing - BookShoppingDsl.java [cd-acceptance-testing_test]
                                                                                                      🔨 🌃 AddBookToBasketTest 🕶 🕨 🇯 🕠 🔲 Git: 🏑 🗸 🚫 🖒 📭 🖭 🔾
 acceptance-testing src test Java com cd acceptance dsl 3 BookShoppingDsl 6 checkOut
                      🕀 🛨 💠 — 🔞 BookDepositoryBookShopDriver.java × 🔞 BookShopDriver.java × 🔞 BookShopDriver.java × 🔞 MyLocalBookStoreBookShopDriver.java ×
  ▼ 📭 acceptance-testing ~/dev/acceptance
                                          package com.cd.acceptance.dsl;
   ▶ 📜 .gradle
   ▶ ■ .idea
                                           import com.cd.acceptance.dsl.drivers.BookShopDrivers;
    build 📂
    ▶ ■ gradle
                                           public class BookShoppingDsl
    ▼ list src
      ▶ P main
                                              private final BookShopDrivers driver;
       ▼ test
        acceptance
                                              public BookShoppingDsl(BookShopDrivers drivers)
          ▼ D com
             ▼ 🖿 cd
                                                  this.driver = drivers;
               ▼ 🖿 acceptance
                 examples
                      ExampleAma
                                              public void searchForBook(String... args)
                      G ExampleMyLo
                                                  Params params = new Params(args);
                 ▶ Di utils
                                                  String title = params.Optional( name: "title", defaultValue: "Continuous Delivery");
           ▼ 🛅 com
                                                  driver.searchForBook(title);
             ▼ Da cd
               V acceptance
                 ▼ Di dsl
                   ▶ 🖿 drivers
                                              public void selectBook(String... args)
                                                  Params params = new Params(args);
                       Channel
                                                  String author = params.Optional( name: "author", defaultValue: "David Farley");
                      Channels
                      CucumberDsl
                                                  driver.selectBook(author);
                      C Dsl
                       TestAnnotatio
                                              public void addSelectedItemToShoppingBasket() { driver.addSelectedItemToShoppingBasket(); }
          ▶ ☐ feature
        ► F resources
                                               public void assertItemListedInShoppingBasket(String... args)
       w build.gradle
                                                  Params params = new Params(args);
      ■ gradlew
                                                  String item = params.Optional( name: "item", defaultValue: "Continuous Delivery");
       gradlew.bat
      run-selenium.sh
                                                   driver.assertListedInShoppingBasket(item);
      a settings.gradle
  ▶ Ilili External Libraries
    Scratches and Consoles
                                              public void checkOut(String... args)
                                                  Params params = new Params(args);
                                                  String item = params.Optional( name: "item", defaultValue: "Continuous Delivery");
                                                  String price = params.Optional( name: "price", defaultValue: "£10.00");
                                                  Card card = parseCard(params.Optional( name: "card", defaultValue: "1234 5678 9101 0001 12/23 007"));
                                                  driver.checkOut(item, price, card);
                                              public void assertItemPurchased(String... args)
                                                  Daname naname - nam Daname(anne)
☐ IntelfiJ IDEA 2020.2.1 available: // Update... (yesterday 19:58)
                                                                                                                                      48:82 LF UTF-8 4 spaces 🏲 master 🚡 👮
```



Continuous

```
. .
                                                       acceptance-testing - BookShoppingDsl.java [cd-acceptance-testing_test]
                                                                                                      🔨 🌃 AddBookToBasketTest 🕶 🕨 🇯 🕠 🔲 Git: 🏑 🗸 🚫 🖒 📭 🖭 🔾
 acceptance-testing src test Java com cd acceptance dsl 3 BookShoppingDsl 6 checkOut
                      🕀 🛨 💠 — 🔞 BookDepositoryBookShopDriver.java × 🔞 BookShopDriver.java × 🔞 BookShopDriver.java × 🔞 MyLocalBookStoreBookShopDriver.java ×
  ▼ 📭 acceptance-testing ~/dev/acceptance
                                          package com.cd.acceptance.dsl;
   ▶ 📜 .gradle
   ▶ ■ .idea
                                           import com.cd.acceptance.dsl.drivers.BookShopDrivers;
    build 📂
    ▶ ■ gradle
                                           public class BookShoppingDsl
    ▼ list src
      ▶ P main
                                              private final BookShopDrivers driver;
       ▼ test
        acceptance
                                              public BookShoppingDsl(BookShopDrivers drivers)
          ▼ D com
             ▼ 🖿 cd
                                                  this.driver = drivers;
               ▼ 🖿 acceptance
                 examples
                      ExampleAma
                                              public void searchForBook(String... args)
                      G ExampleMyLo
                                                  Params params = new Params(args);
                 ▶ Di utils
                                                  String title = params.Optional( name: "title", defaultValue: "Continuous Delivery");
           ▼ 🛅 com
                                                  driver.searchForBook(title);
             ▼ Da cd
               V acceptance
                 ▼ Di dsl
                   ▶ 🖿 drivers
                                              public void selectBook(String... args)
                                                  Params params = new Params(args);
                       Channel
                                                  String author = params.Optional( name: "author", defaultValue: "David Farley");
                      Channels
                      CucumberDsl
                                                  driver.selectBook(author);
                      C Dsl
                       TestAnnotatio
                                              public void addSelectedItemToShoppingBasket() { driver.addSelectedItemToShoppingBasket(); }
          ▶ ☐ feature
        ► F resources
                                               public void assertItemListedInShoppingBasket(String... args)
       w build.gradle
                                                  Params params = new Params(args);
      ■ gradlew
                                                  String item = params.Optional( name: "item", defaultValue: "Continuous Delivery");
       gradlew.bat
      run-selenium.sh
                                                   driver.assertListedInShoppingBasket(item);
      a settings.gradle
  ▶ Ilili External Libraries
    Scratches and Consoles
                                              public void checkOut(String... args)
                                                  Params params = new Params(args);
                                                  String item = params.Optional( name: "item", defaultValue: "Continuous Delivery");
                                                  String price = params.Optional( name: "price", defaultValue: "£10.00");
                                                  Card card = parseCard(params.Optional( name: "card", defaultValue: "1234 5678 9101 0001 12/23 007"));
                                                  driver.checkOut(item, price, card);
                                              public void assertItemPurchased(String... args)
                                                  Daname naname - nam Daname(anne)
☐ IntelfiJ IDEA 2020.2.1 available: // Update... (yesterday 19:58)
                                                                                                                                      48:82 LF UTF-8 4 spaces 🏲 master 🚡 👮
```



Continuous

0010 00 00 01 01 0111011001100101010100010111010100 1011101010101010 10016010181000 00 00010010010011 801610010101001101010101011 0100104 001000 01 010010101001 0 001011001801001001081000100101 1010 | 110 10 100 010101000100 000100 00100111001 | 0010101001 01010001 310-1901 0100 00 1110 100 1 0101101000101 1010100010111010101010101 88108 801880 01 010010101001 0 00101100108108100108108010010101 10 | 1101 | 100 | 201 | 201 | 100 | 100 | 100 | 100 | 100 | 100 | 101 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 1



0010 00 00 01 01 0111011001100101010100010111010100 1011101010101010 10016010181000 00 00010010010011 801610010101001101010101011 0100104 001000 01 010010101001 0 001011001801001001081000100101 1010 | 110 10 100 010101000100 000100 00100111001 | 0010101001 01010001 310-1901 0100 00 1110 100 1 0101101000101 1010100010111010101010101 88108 801880 01 010010101001 0 00101100108108100108108010010101 10 | 1101 | 100 | 201 | 201 | 100 | 100 | 100 | 100 | 100 | 100 | 101 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 11 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 1





0010 00 00 01 01 0111011001100101010100010111010100 101110101010101010 88918818 | 188188 | 1890181 | 181188 | 1801818 | 1808 | 18118 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1818 | 1 01 11100 0010010 0 00010010 01000100 001 1001010 1010 00110101010011 1010 - 110 10 10010101000100 000100 00100111001 0010101001 010100011 1010 10 10 10 010100 10010001 010010011100101010101010101010101010101 81881881 0188 00 11101100 1 0101101000101 10101000101110101010101110 01 11100 0010010 01000100100100100100 001 1001010 1010 00110101010011 08 401 0100 0100011101100110010 101000101110101000101110101010 0110



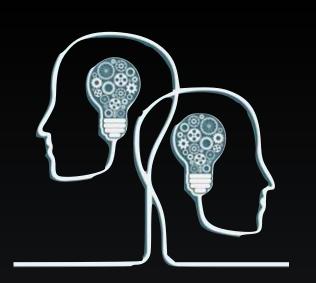
DEPLOY





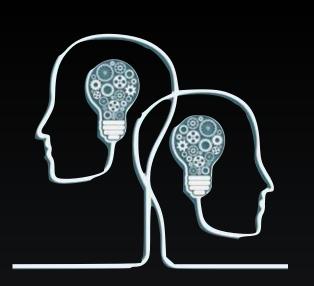






Engineering *IS* About & Discovery & Discovery





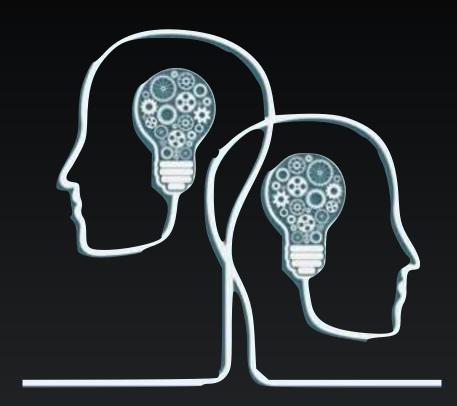
Development & S* About & Discovery Exploration & Discovery



The Stuff That Works



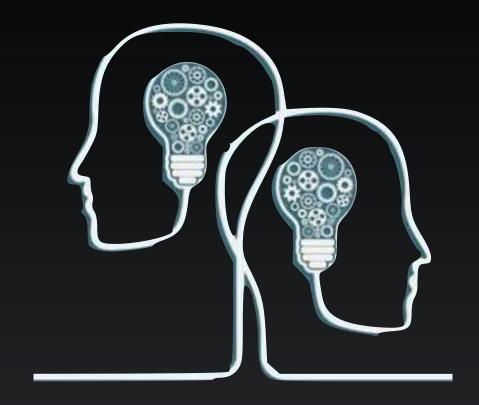
The Stuff That Works

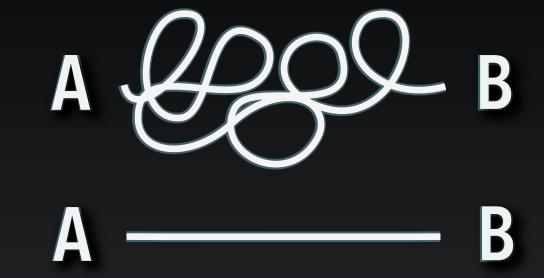


Optimise for Learning



The Stuff That Works





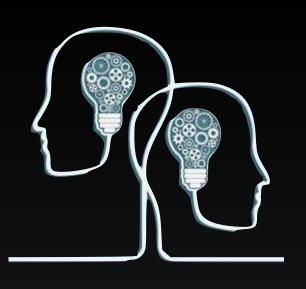
Optimise for Learning

Optimise for Managing Complexity





Optimise for Learning 💮

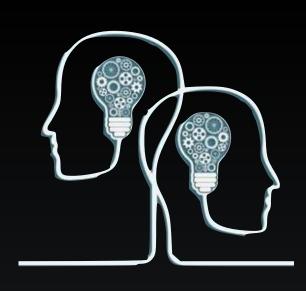


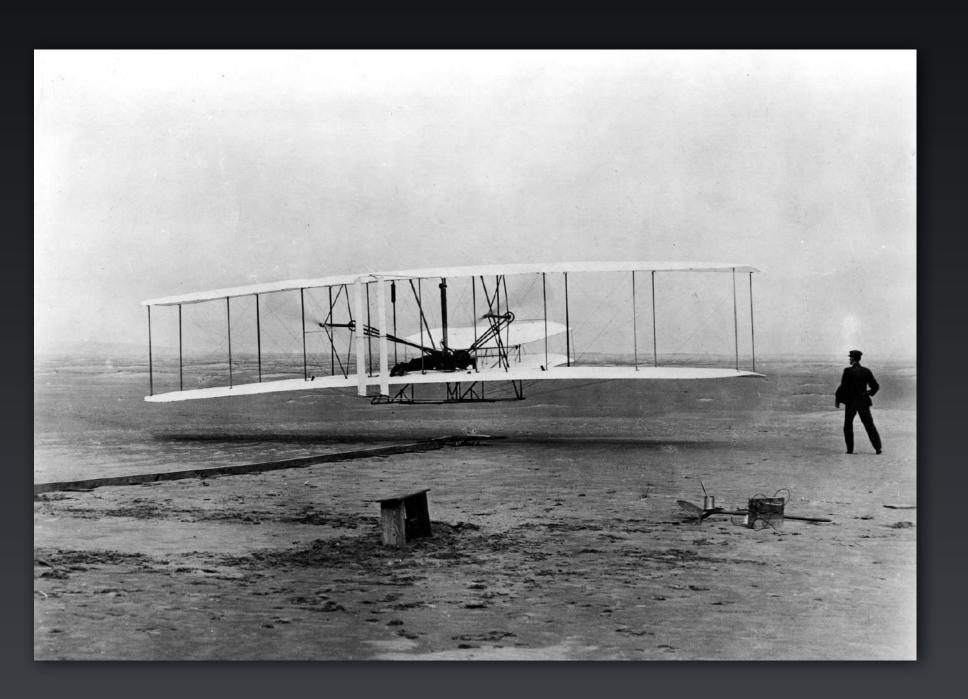
Iterative Experimental

Feedback Driven Incremental Empirical



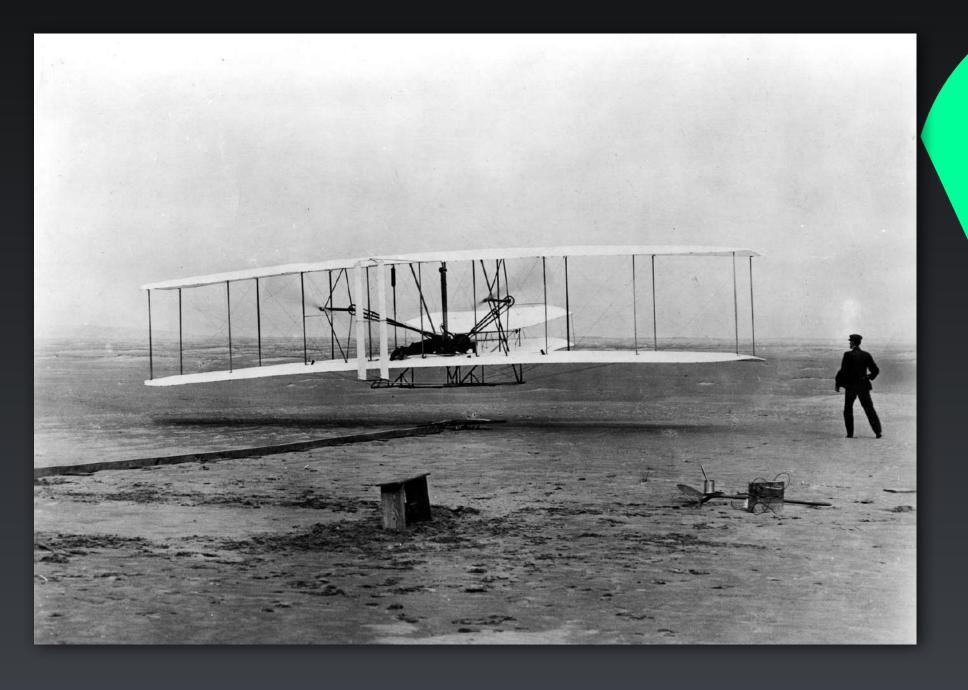








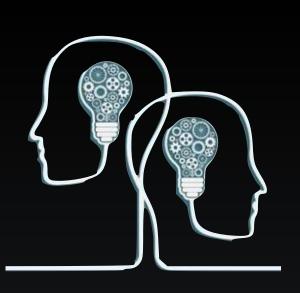




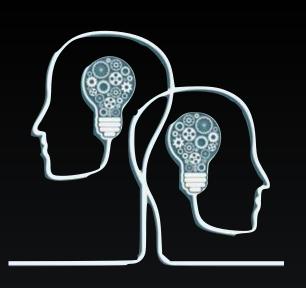






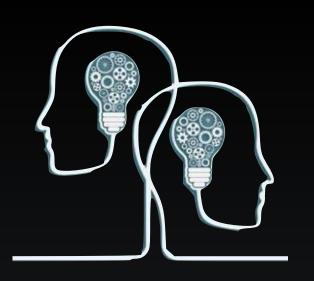








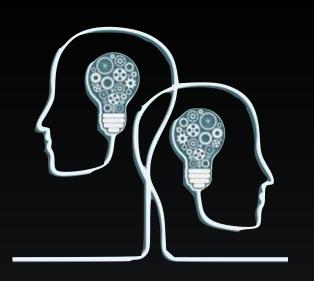










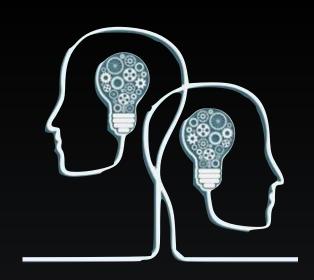






Fitness Function



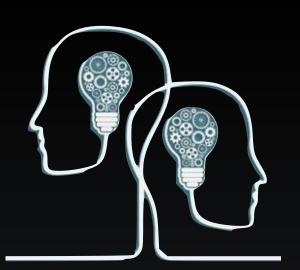




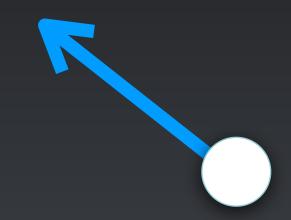


Fitness Function

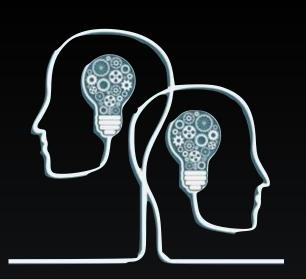




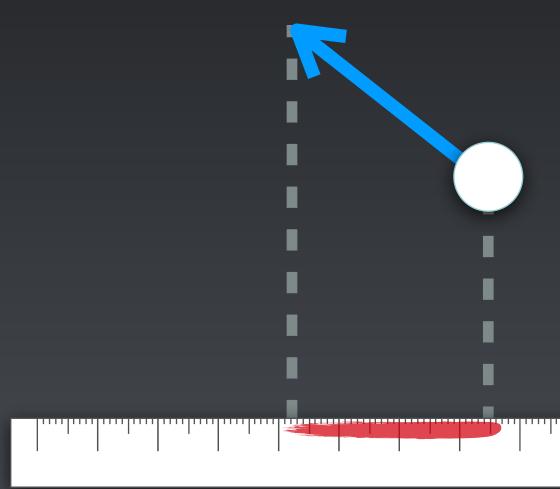




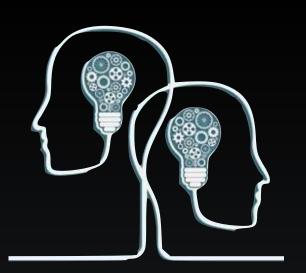






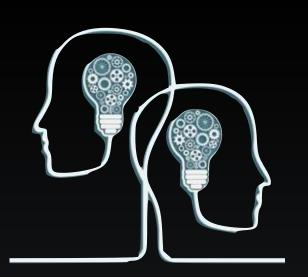








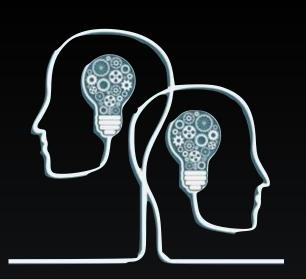


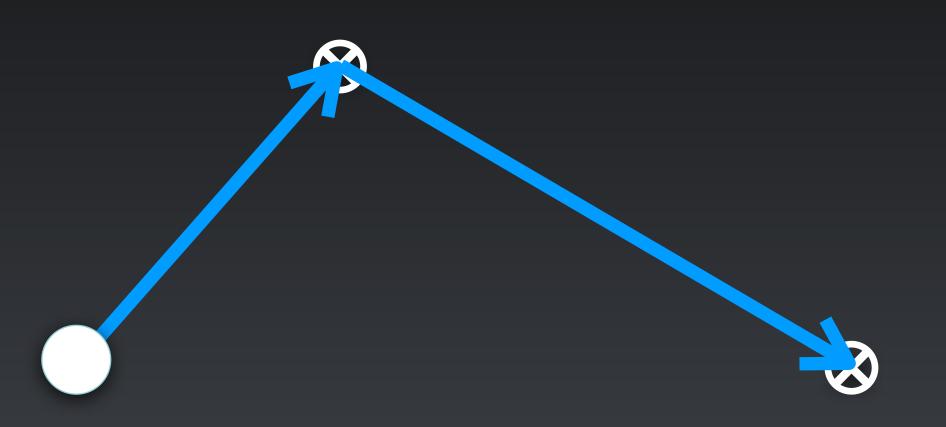






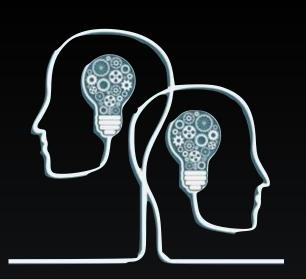


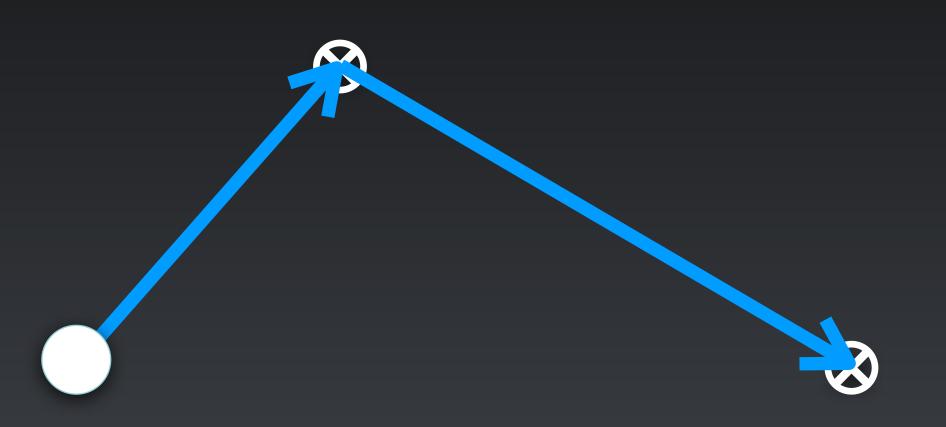






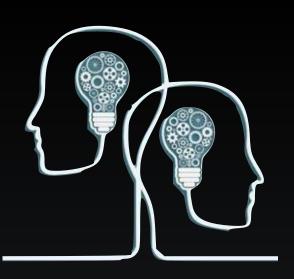


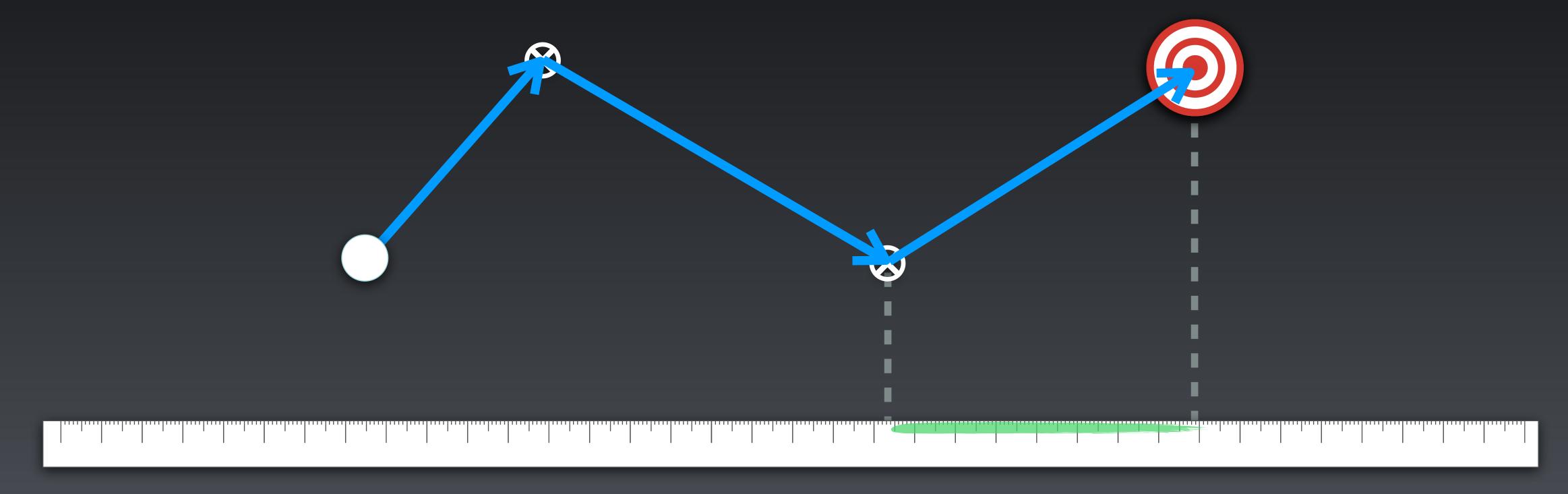






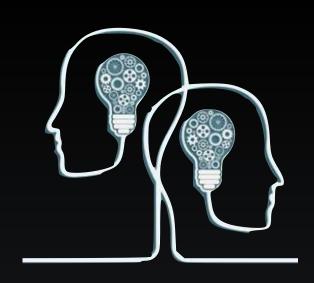


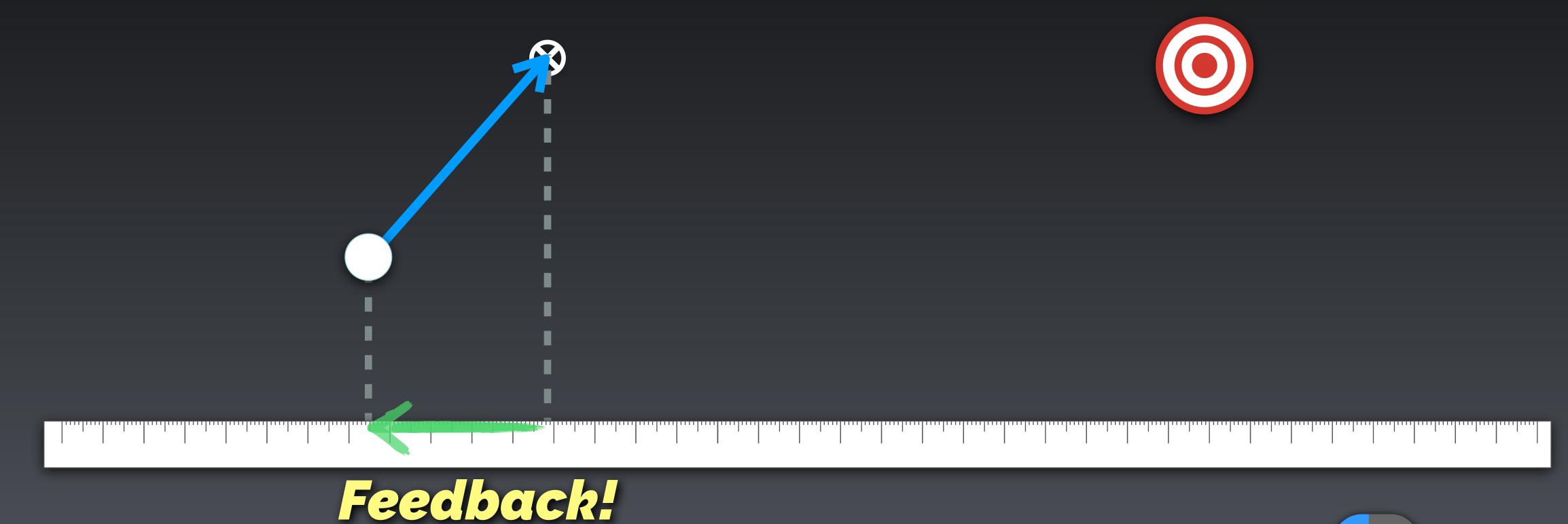




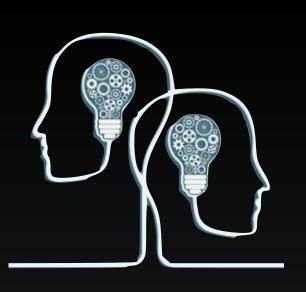
















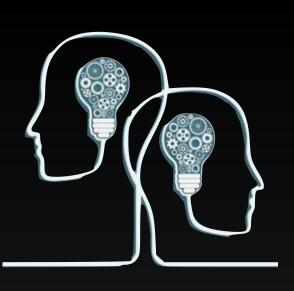






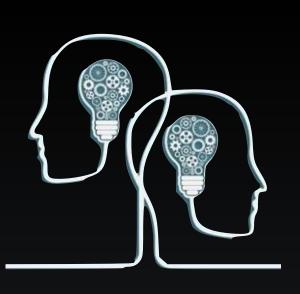






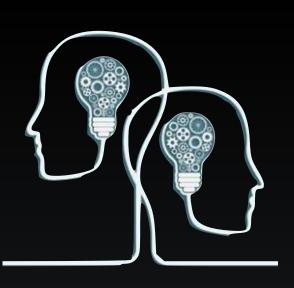








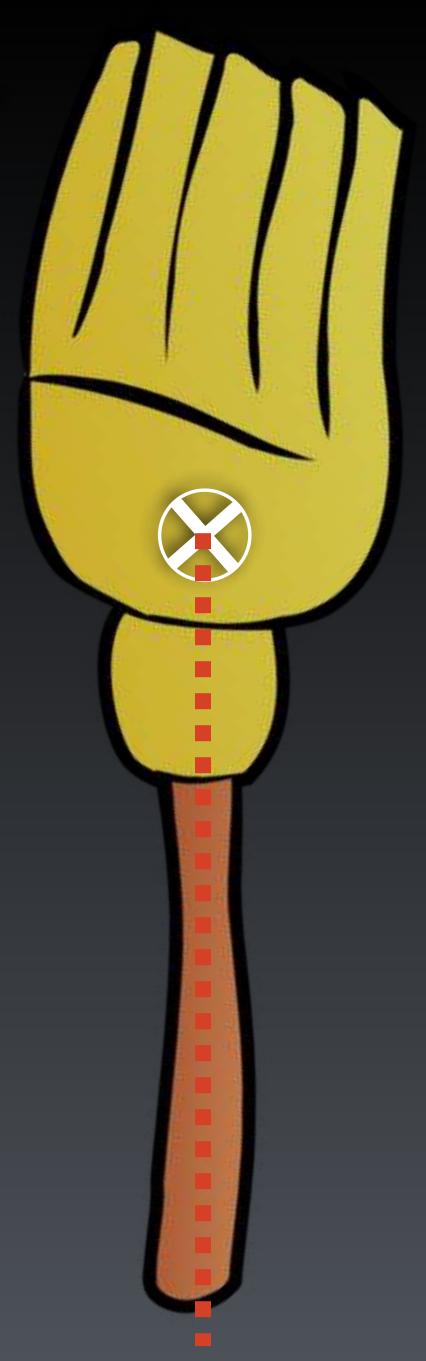


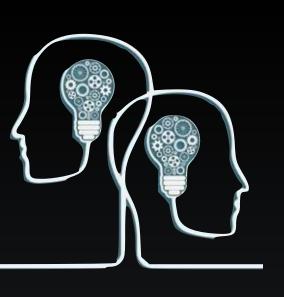






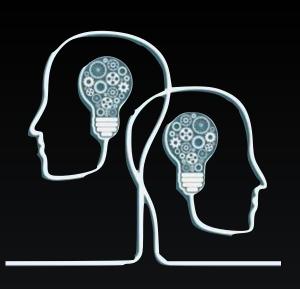
Feedback //



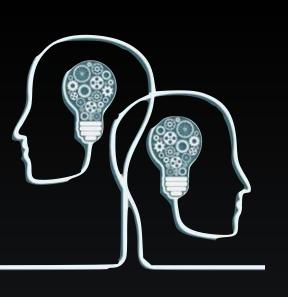


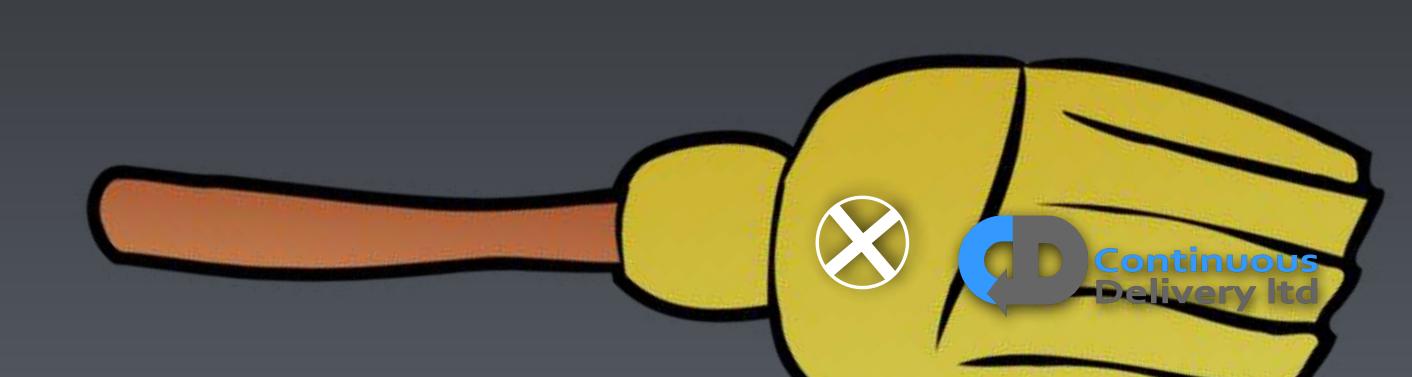


reed back







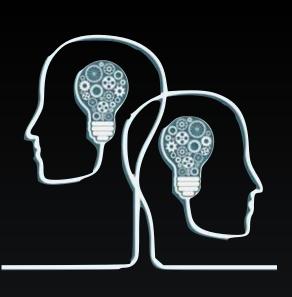






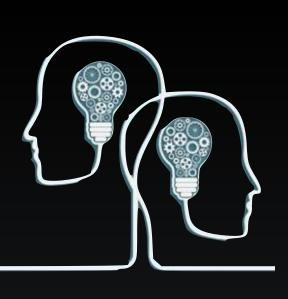


reeaback //





reeaback /



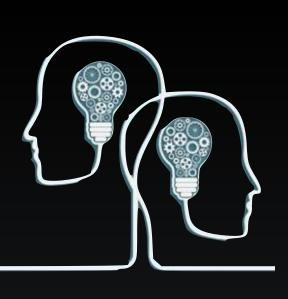


Feedback III





reeaback /

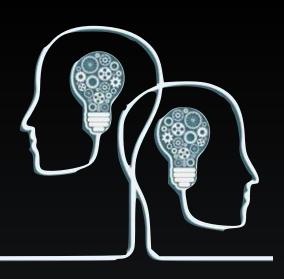




Feedback III





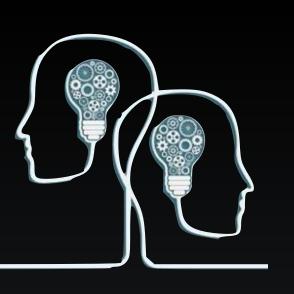






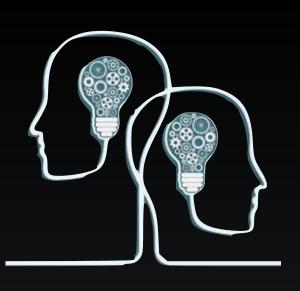


ncremental



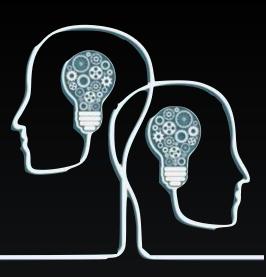


ncremental





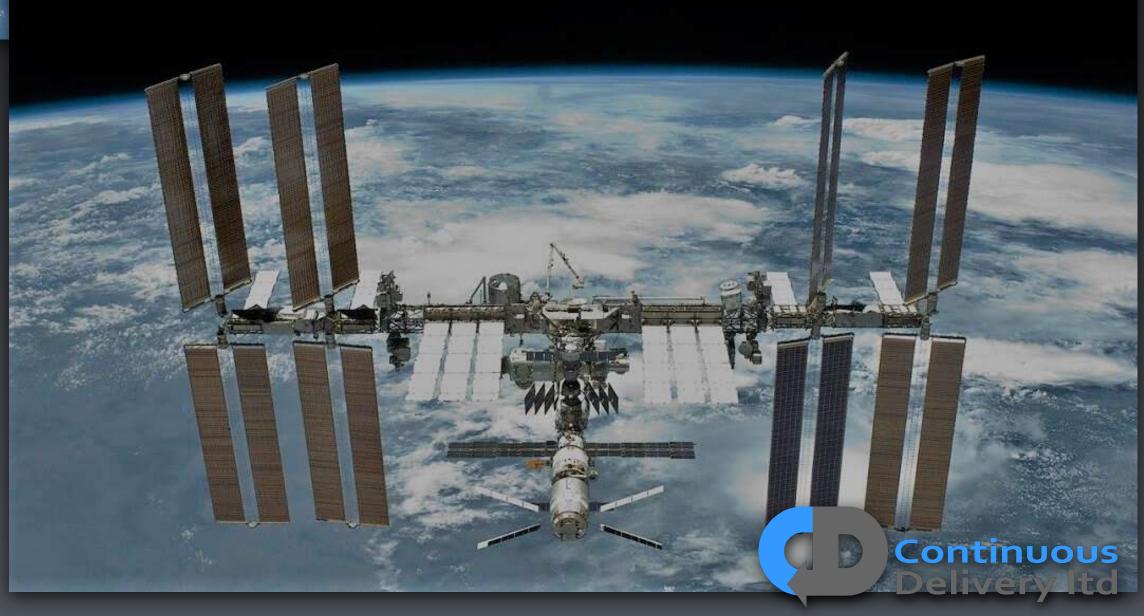


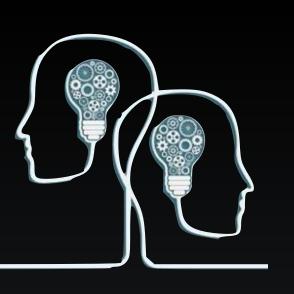




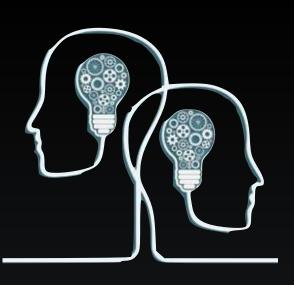






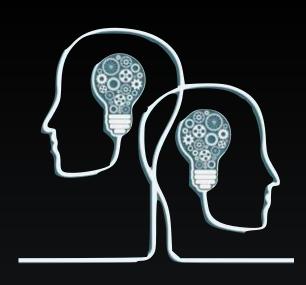








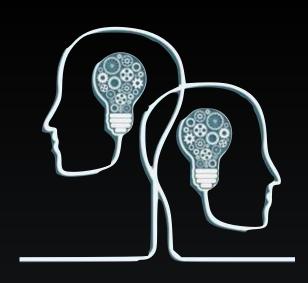






No 3rd Party Apps No GPS No Video No Facetime





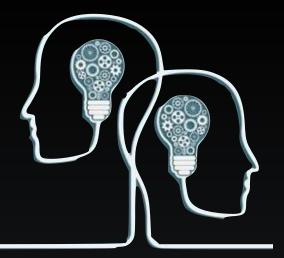




No 3rd Party Apps
No GPS
No Video
No Facetime

No 4G No Siri No Face Recognition No Apple Pay







No 3rd Party Apps
No GPS
No Video
No Facetime



No 4G No Siri No Face Recognition No Apple Pay



What Comes Next?





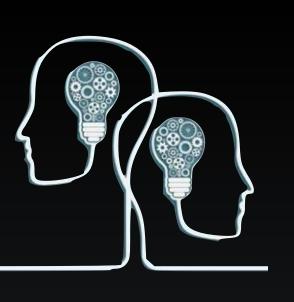


Scary

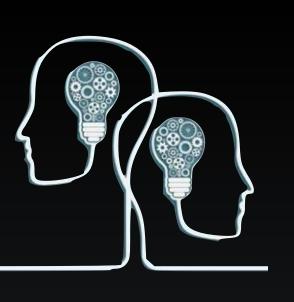


Experimental How Else Can We Make Progress?





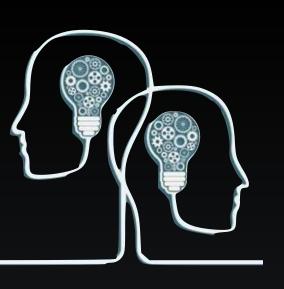






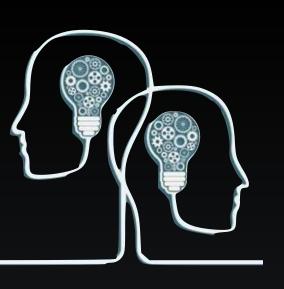
Experimental Controlling the Variables!

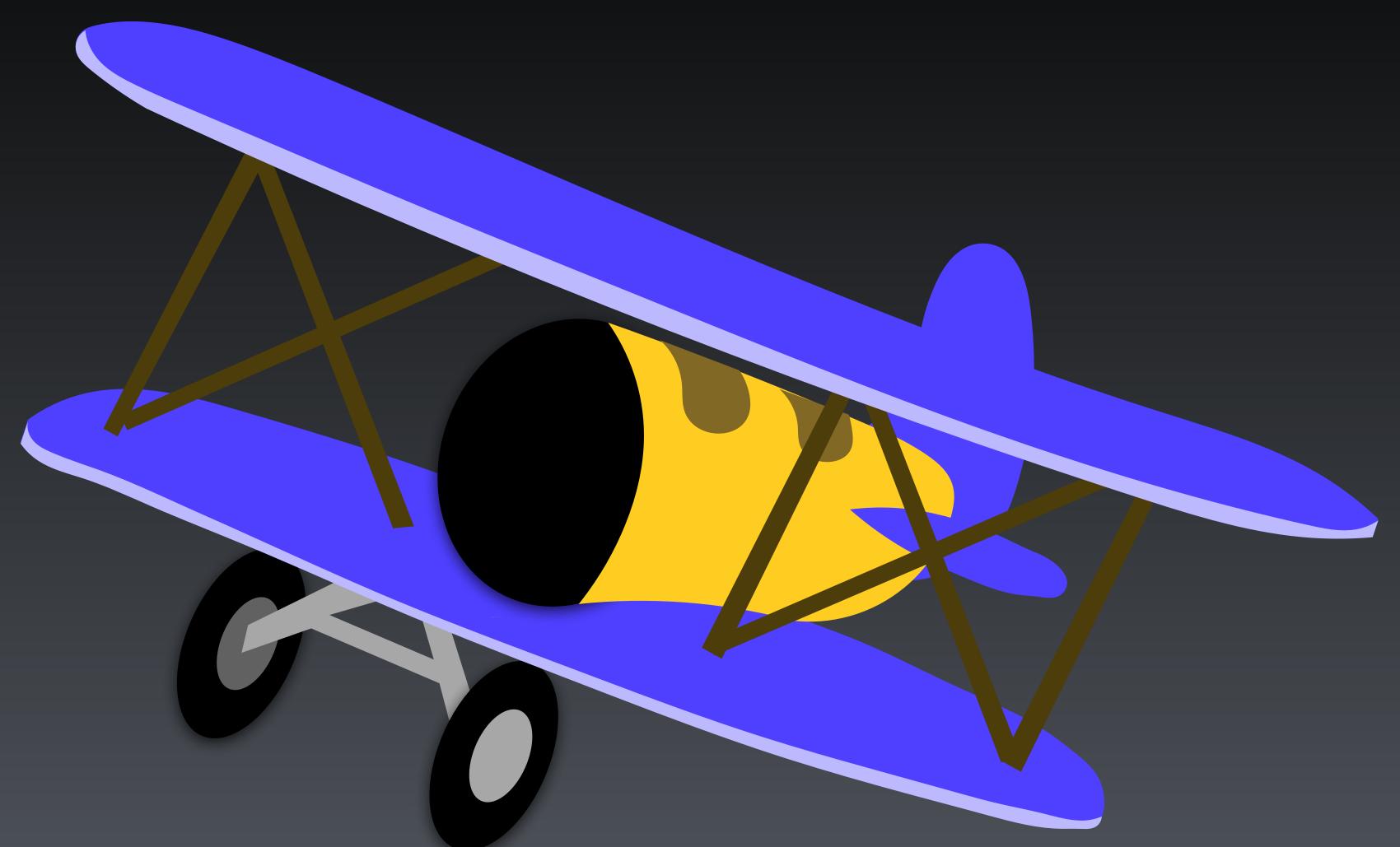




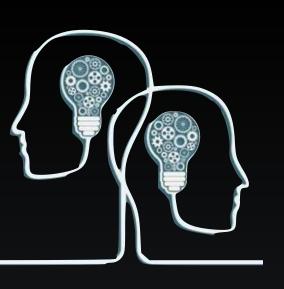


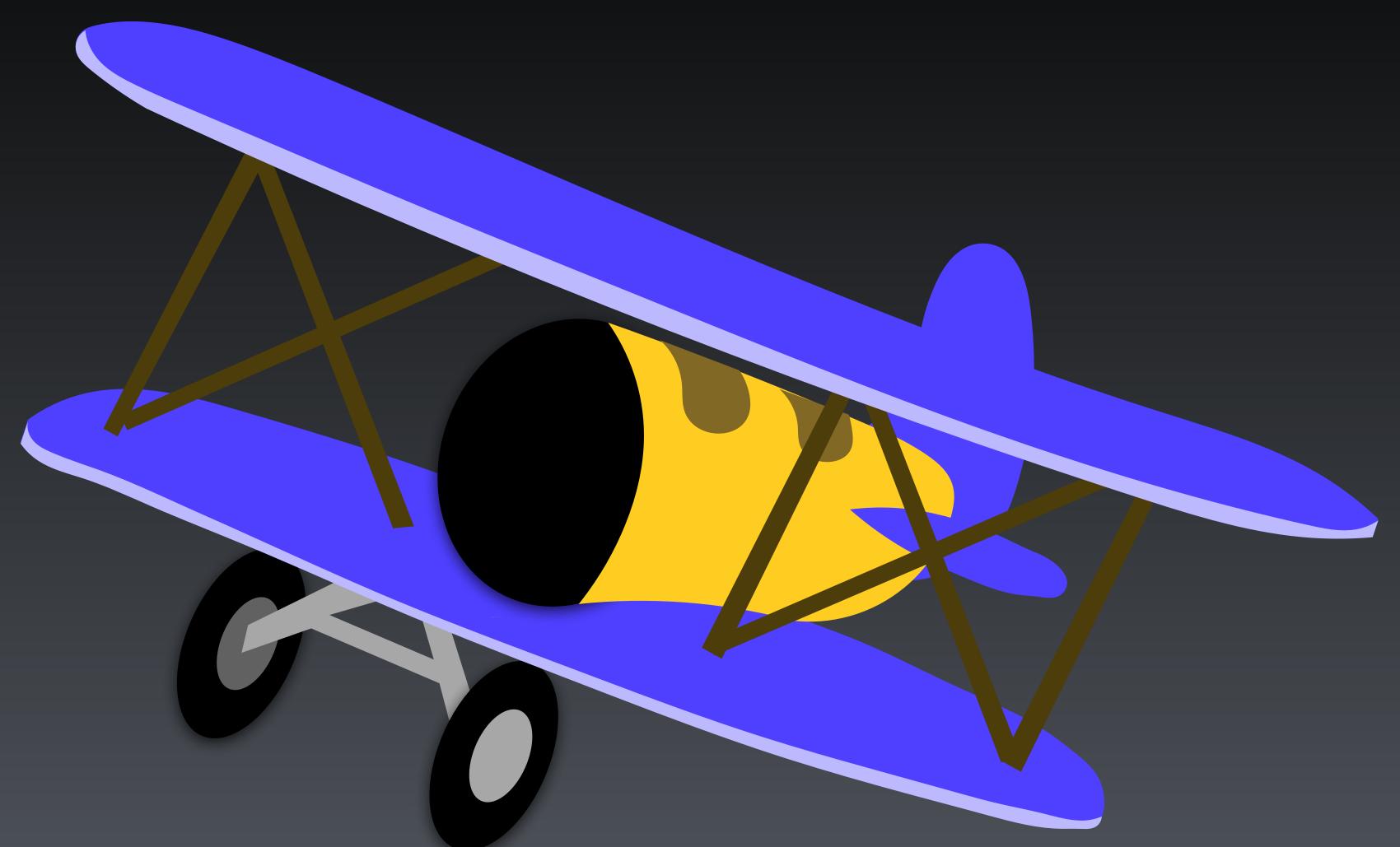




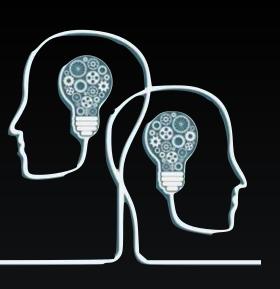






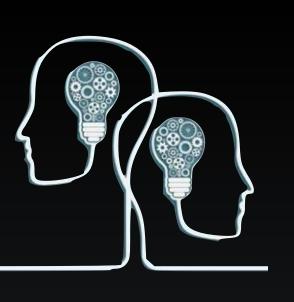




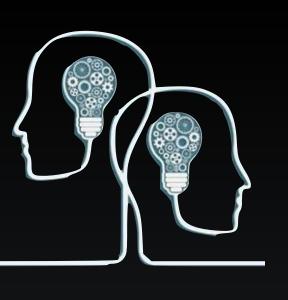


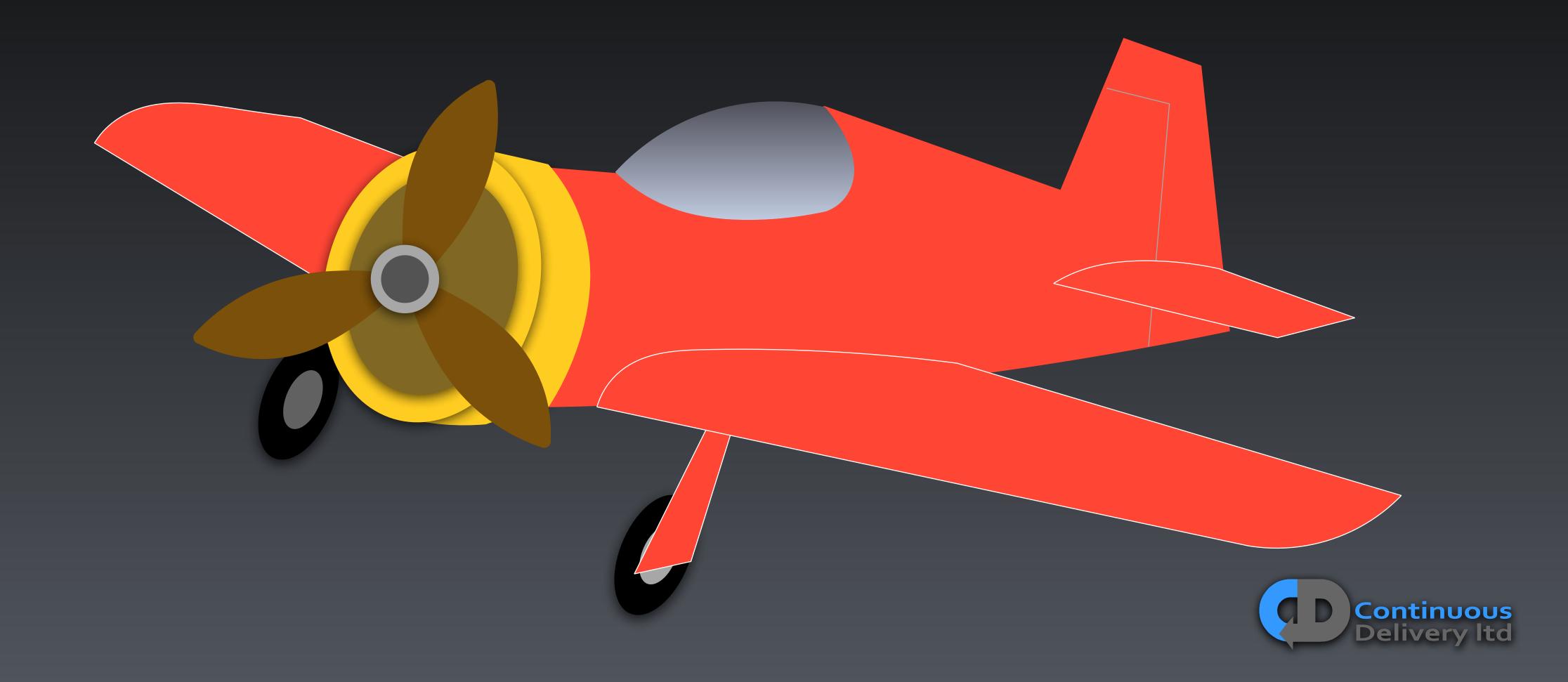


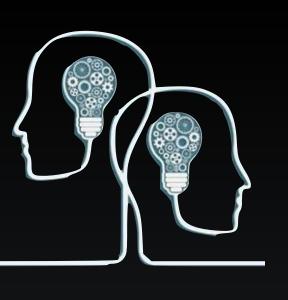


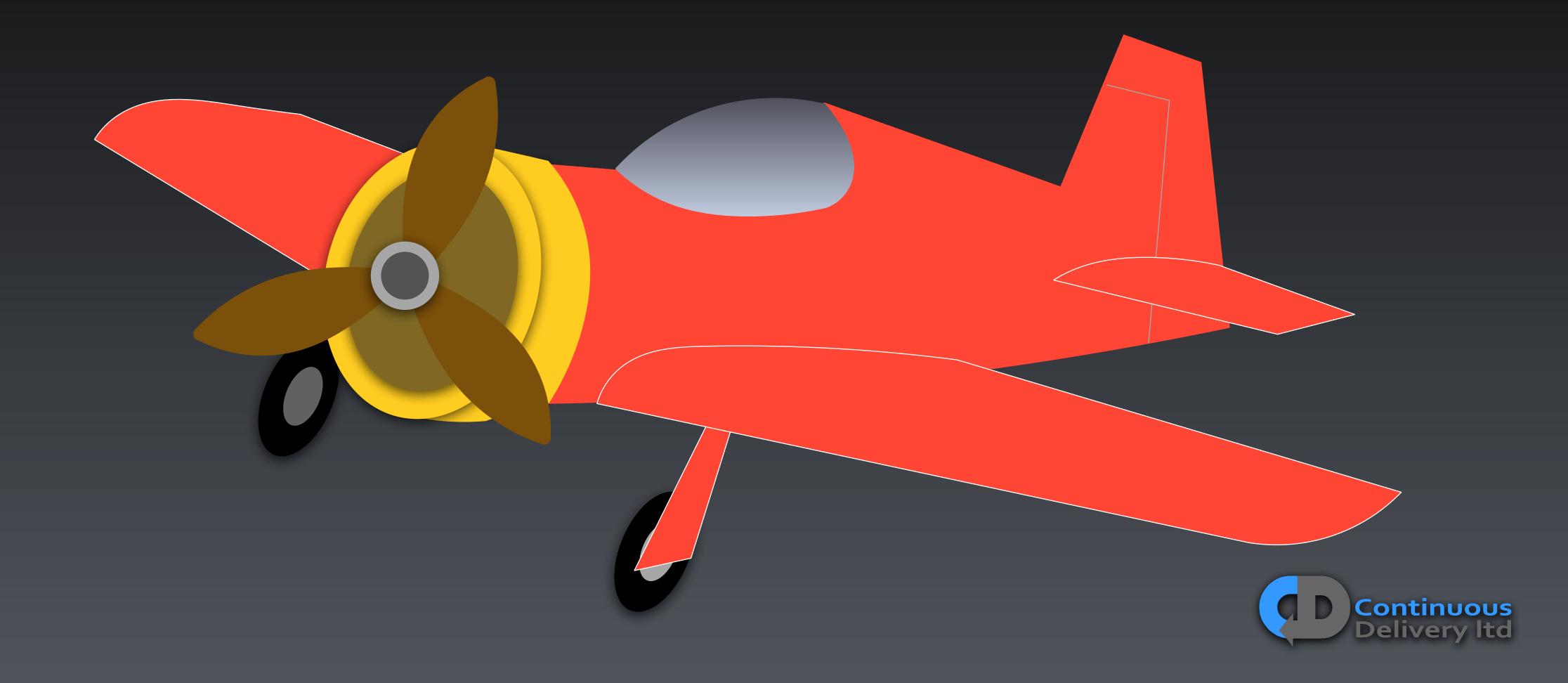


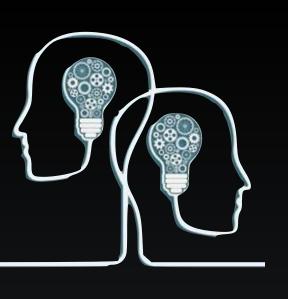


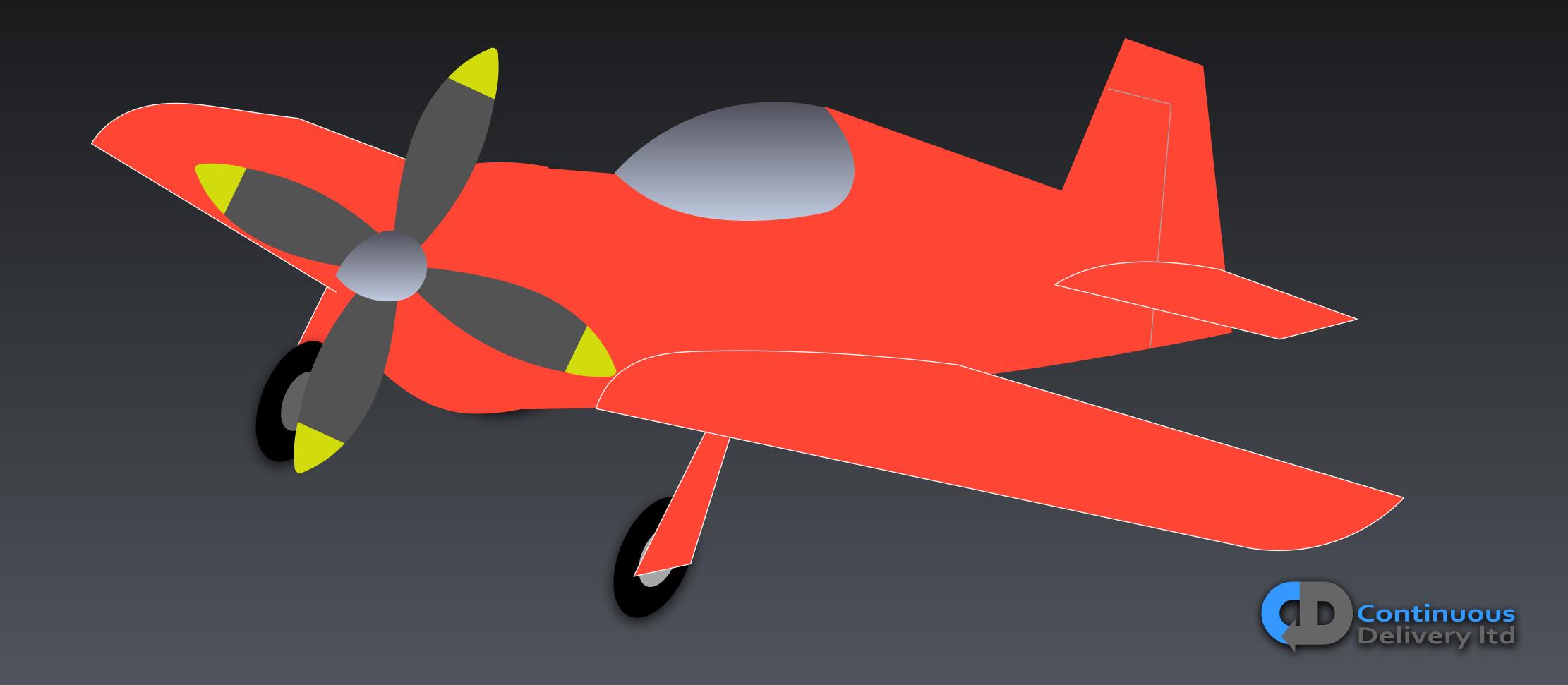












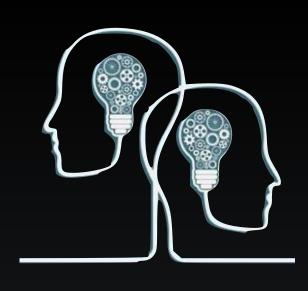


Empirical



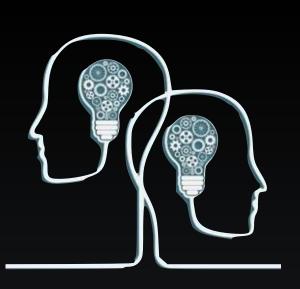


Empirica.



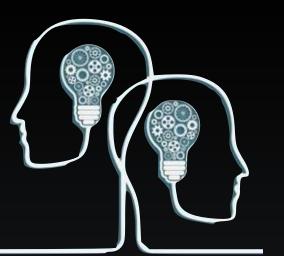






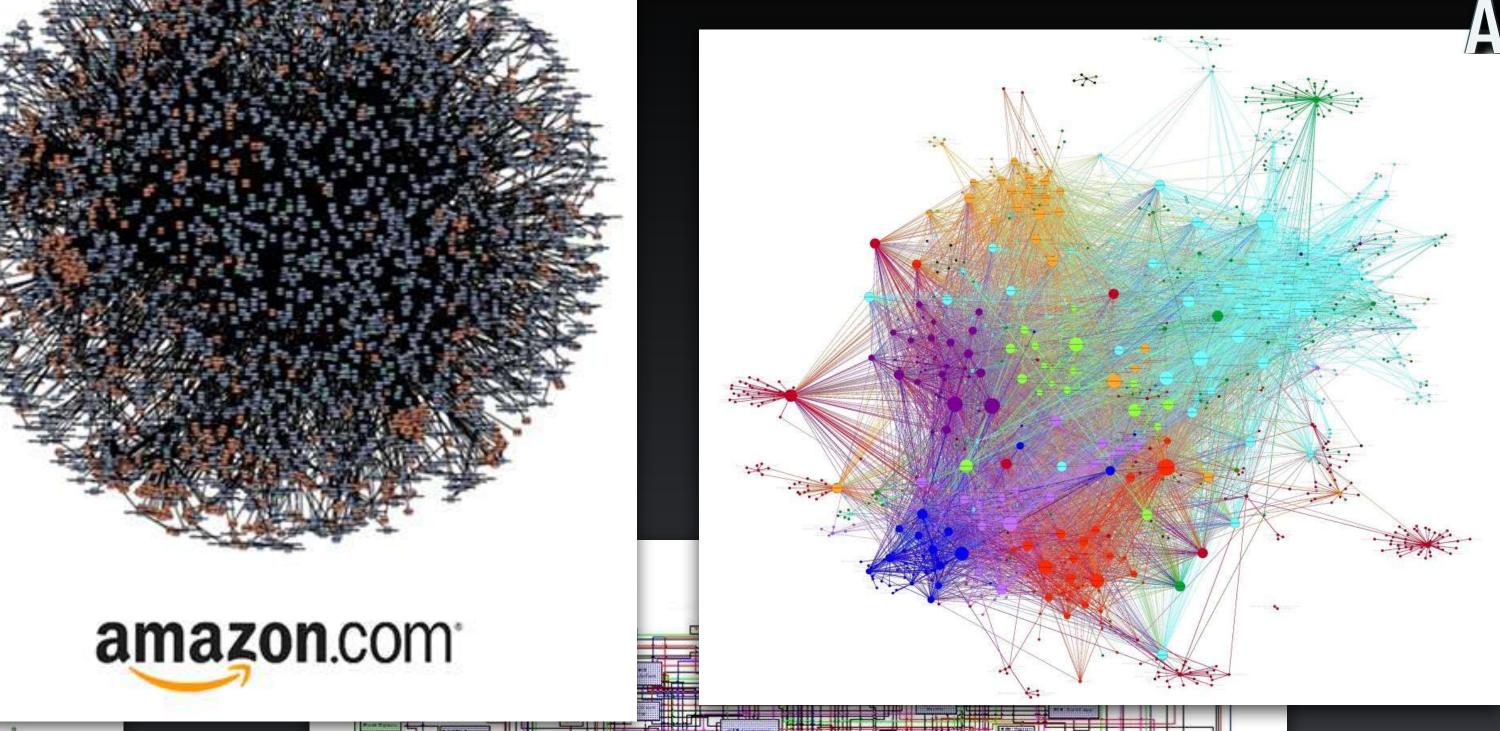


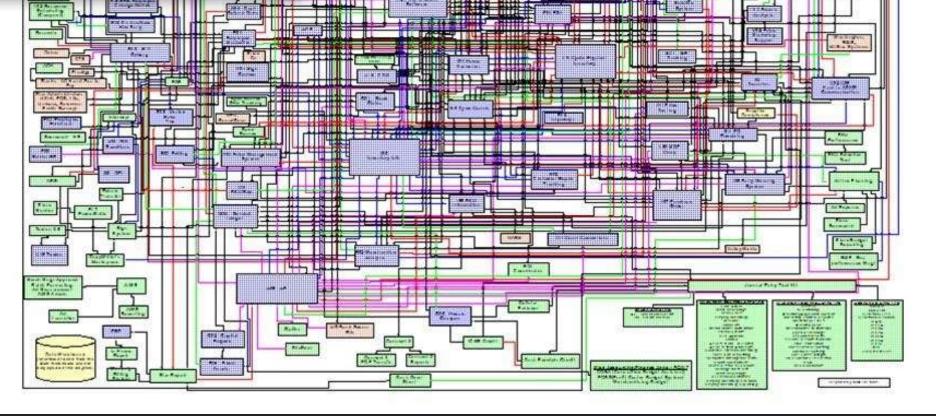
Empirical.







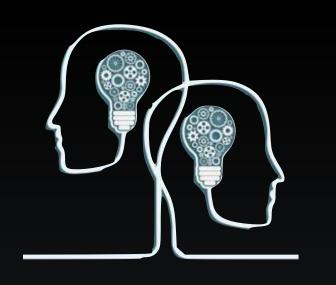


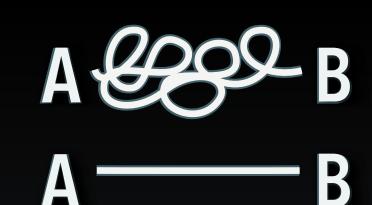






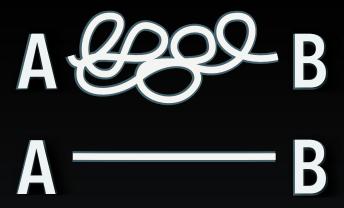






Software Development is about Learning & Managing Complexity





Managing Complexity





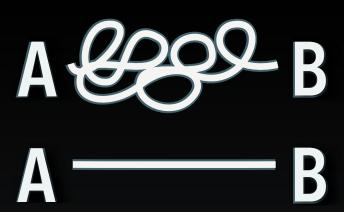
Managing Complexity

Modularity Cohesion
Separation of Concerns
Abstraction Coupling



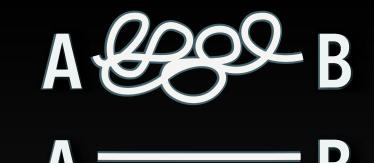


Modularity





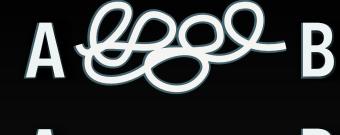
Modularity





Modularity



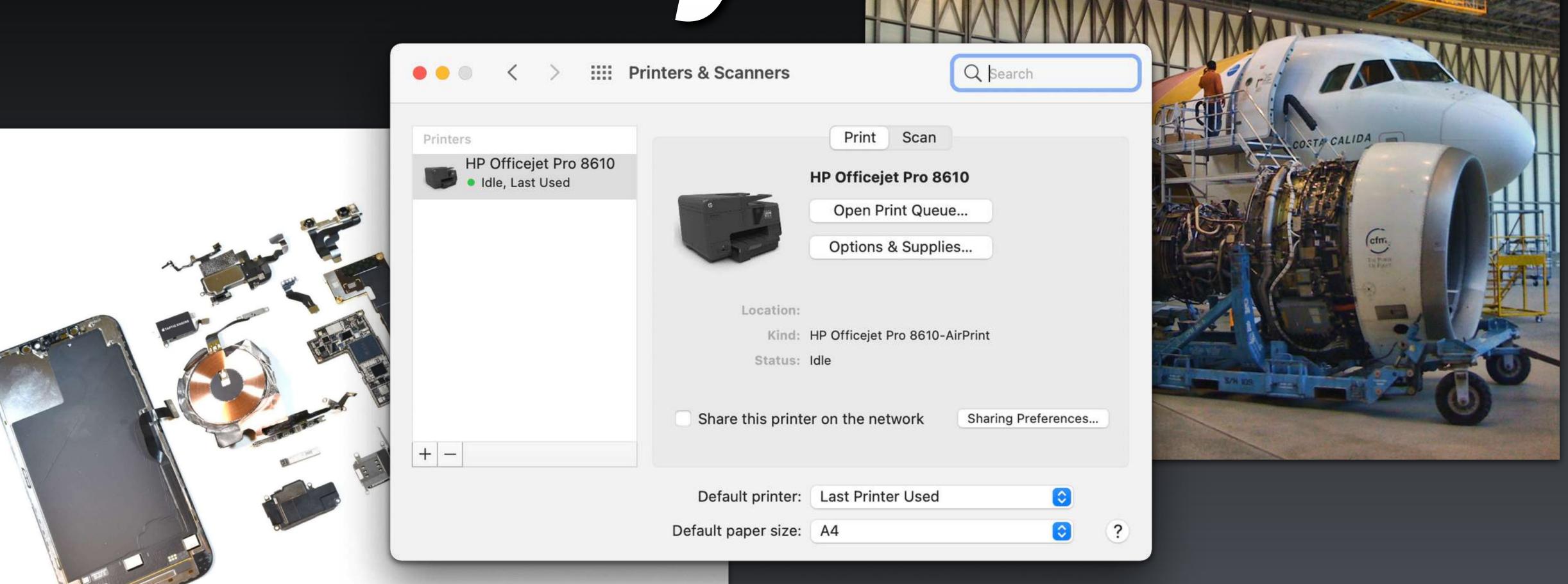


A ——— E





A B B





Continuous

Delivery Itd

```
* ecm_nss_non_ported_ipv4_process()
      * Process a protocol that does not have port based identifiers
     unsigned int ecm_nss_non_ported_ipv4_process(struct net_device *out_dev, struct net_device *out_dev_nat,
                                 struct net_device *in_dev, struct net_device *in_dev_nat,
6
                                 uint8_t *src_node_addr, uint8_t *src_node_addr_nat,
                                 uint8_t *dest_node_addr, uint8_t *dest_node_addr_nat,
                                 bool can_accel, bool is_routed, bool is_l2_encap, struct sk_buff *skb,
9
                                 struct ecm_tracker_ip_header *ip_hdr,
10
                                 struct nf_conn *ct, ecm_tracker_sender_type_t sender, ecm_db_direction_t ecm_dir,
11
12
                                 struct nf_conntrack_tuple *orig_tuple, struct nf_conntrack_tuple *reply_tuple,
                                 ip_addr_t ip_src_addr, ip_addr_t ip_dest_addr, ip_addr_t ip_src_addr_nat, ip_addr_t
13
     ip_dest_addr_nat)
14 ▼
         struct ecm_db_connection_instance *ci;
15
16
         int protocol;
         int src_port;
17
18
         int src_port_nat;
19
         int dest_port;
20
         int dest_port_nat;
21
         ip_addr_t match_addr;
         struct ecm_classifier_instance *assignments[ECM_CLASSIFIER_TYPES];
22
23
         int aci_index;
         int assignment_count;
24
         ecm_db_timer_group_t ci_orig_timer_group;
25
         struct ecm_classifier_process_response prevalent_pr;
26
27
         DEBUG_TRACE("Non-ported protocol src: " ECM_IP_ADDR_DOT_FMT ", dest: " ECM_IP_ADDR_DOT_FMT "\n",
28
                     ECM_IP_ADDR_TO_DOT(ip_src_addr), ECM_IP_ADDR_TO_DOT(ip_dest_addr));
29
30
```

Continuous Delivery Itd

Modularity

of the state of the

```
if (unlikely(!ci)) {
60 ▼
              struct ecm_db_mapping_instance *src_
61
              struct ecm_db_mapping_instance *dest_mi;
62
              struct ecm_db_mapping_instance *src_nat_mi;
63
              struct ecm_db_mapping_instance *dest_nat_mi;
64
              struct ecm_db_node_instance *src_ni;
65
              struct ecm_db_node_instance *dest_ni;
66
              struct ecm_db_node_instance *src_nat_ni;
67
              struct ecm_db_node_instance *dest_nat_ni;
68
              struct ecm_classifier_default_instance *dci;
69
              struct ecm_front_end_connection_instance *feci;
70
              struct ecm_db_connection_instance *nci;
71
              ecm_classifier_type_t classifier_type;
72
              int32_t to_list_first;
73
              struct ecm_db_iface_instance *to_list[ECM_DB_IFACE_HEIRARCHY_MAX];
74
              int32_t to_nat_list_first;
75
              struct ecm_db_iface_instance *to_nat_list[ECM_DB_IFACE_HEIRARCHY_MAX];
76
              int32_t from_list_first;
77
              struct ecm_db_iface_instance *from_list[ECM_DB_IFACE_HEIRARCHY_MAX];
78
             int32_t from_nat_list_first;
79
              struct ecm_db_iface_instance *from_nat_list[ECM_DB_IFACE_HEIRARCHY_MAX];
80
81
              DEBUG_INFO("New connection from " ECM_IP_ADDR_DOT_FMT " to " ECM_IP_ADDR_DOT_FMT "\n",
82
                         ECM_IP_ADDR_TO_DOT(ip_src_addr), ECM_IP_ADDR_TO_DOT(ip_dest_addr));
83
84
85
86
87
              spin_lock_bh(&ecm_nss_ipv4_lock);
88
                (ecm_nss_ipv4_terminate_pending) -
89 ▼
                 spin unlock bh(&ecm nss inv4 lock):
90
```

422

423

```
394
                  ecm_db_connection_add(nci, feci
395
                                                    c_mi, dest_mi, src_nat_mi, dest_nat_mi,
                           src_ni, dest_ni, src_nat_ni, dest_nat_ni,
396
                           4, protocol, ecm_dir,
397
                          NULL /* final callback */,
398
                          ecm_nss_non_ported_ipv4_connection_defunct_callback,
399
                           tg, is_routed, nci);
400
401
                  spin_unlock_bh(&ecm_nss_ipv4_lock);
402
403
                  ci = nci;
404
                  DEBUG_INFO("%p: New Non-ported pro
405
406 ▲
407
408
409
410
               dci->bas
                                             lassifier_instance
411
412
               ecm_db_map
413
               ecm_db_nou
414
415
               ecm_db_mapping
416
417
               ecm_db_node_
               ecm_db_mappi<del>ng</del>_d
418
419
               ecm_db_node_deref
              feci->deref(feci);
420
421 ▲
```



A LESS B A R

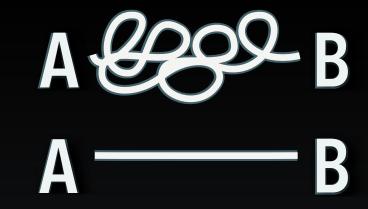
MOGULAITH

```
ed identifiers
      * Process a protocol that does not have por
     unsigned int ecm_nss_non_ported_ipv4_process(struct net_device *out_dev, struct net_device *out_dev_nat,
                                 struct net_device *in_dev, struct net_device *in_dev_nat,
                                 uint8_t *src_node_addr, uint8_t *src_node_addr_nat,
                                 uint8_t *dest_node_addr, uint8_t *dest_node_addr_nat,
                                 bool can_accel, bool is_routed, bool is_l2_encap, struct sk_buff *skb,
                                 struct ecm_tracker_ip_header *ip_hdr,
10
                                 struct nf_conn *ct, ecm_tracker_sender_type_t sender, ecm_db_direction_t ecm_dir,
11
                                 struct nf_conntrack_tuple *orig_tuple, struct nf_conntrack_tuple *reply_tuple,
12
                                 ip_addr_t ip_src_addr, ip_addr_t ip_dest_addr, ip_addr_t ip_src_addr_nat, ip_addr_t
13
     ip_dest_addr_nat)
14 ▼
         struct ecm_db_connection_instance *ci;
15
         int protocol;
16
         int src_port;
17
18
         int src_port_nat;
         int dest_port;
19
         int dest_port_nat;
20
         ip_addr_t match_addr;
21
         struct ecm_classifier_instance *assignments[ECM_CLASSIFIER_TYPES];
22
         int aci_index;
23
         int assignment_count;
24
         ecm_db_timer_group_t ci_orig_timer_group;
25
         struct ecm_classifier_process_response prevalent_pr;
26
27
         DEBUG_TRACE("Non-ported protocol src: " ECM_IP_ADDR_DOT_FMT ", dest: " ECM_IP_ADDR_DOT_FMT "\n",
                     ECM_IP_ADDR_TO_DOT(ip_src_addr), ECM_IP_ADDR_TO_DOT(ip_dest_addr));
29
30
31
```



A B B









```
if (unlikely(!ci)) {
   struct ecm_db_mapping_instance *src_mi;
   struct ecm_db_mapping_instance *dest_mi;
   struct ecm_db_mapping_instance *src_nat_mi;
   struct ecm_db_mapping_instance *dest_nat_mi;
   struct ecm_db_node_instance *src_ni;
   struct ecm_db_node_instance *dest_ni;
   struct ecm_db_node_instance *src_nat_ni;
   struct ecm_db_node_instance *dest_nat_ni;
   struct ecm_classifier_default_instance *dci;
   struct ecm_front_end_connection_instance *feci;
   struct ecm_db_connection_instance *nci;
   ecm_classifier_type_t classifier_type;
   int32_t to_list_first;
   struct ecm_db_iface_instance *to_list[ECM_DB_IFACE_HEIRARCHY_MAX]
   int32_t to_nat_list_first;
   struct ecm_db_iface_instance *to_nat_list[ECM_DB_IFACE_HEIRARCHY_
   int32_t from_list_first;
   struct ecm dh iface instance *from list[FCM DR TFACE HETRARCHY M/
```





```
if (noConnection(ci)) {
    createConnection();
}
```





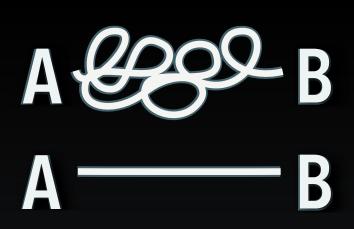


```
if (noConnection(ci)) {
    createConnection();
}
```

```
A B
B
```

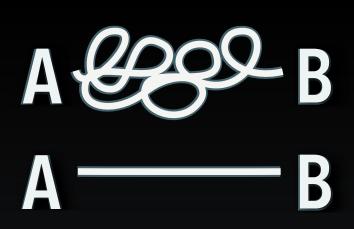






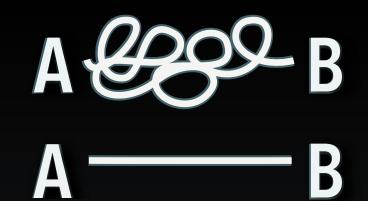
"Good Design is about moving the things that are related closer together, things that aren't related and the things that aren't related further apart." Kent Beck





"Good Design is about moving the things that are related closer together, things that aren't related and the things that aren't related further apart." Kent Beck



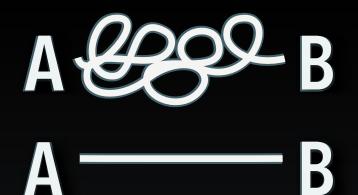


Modularity things that aren't related further apart

Cohesion things that are related closer together



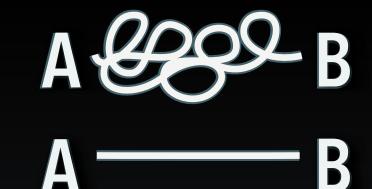


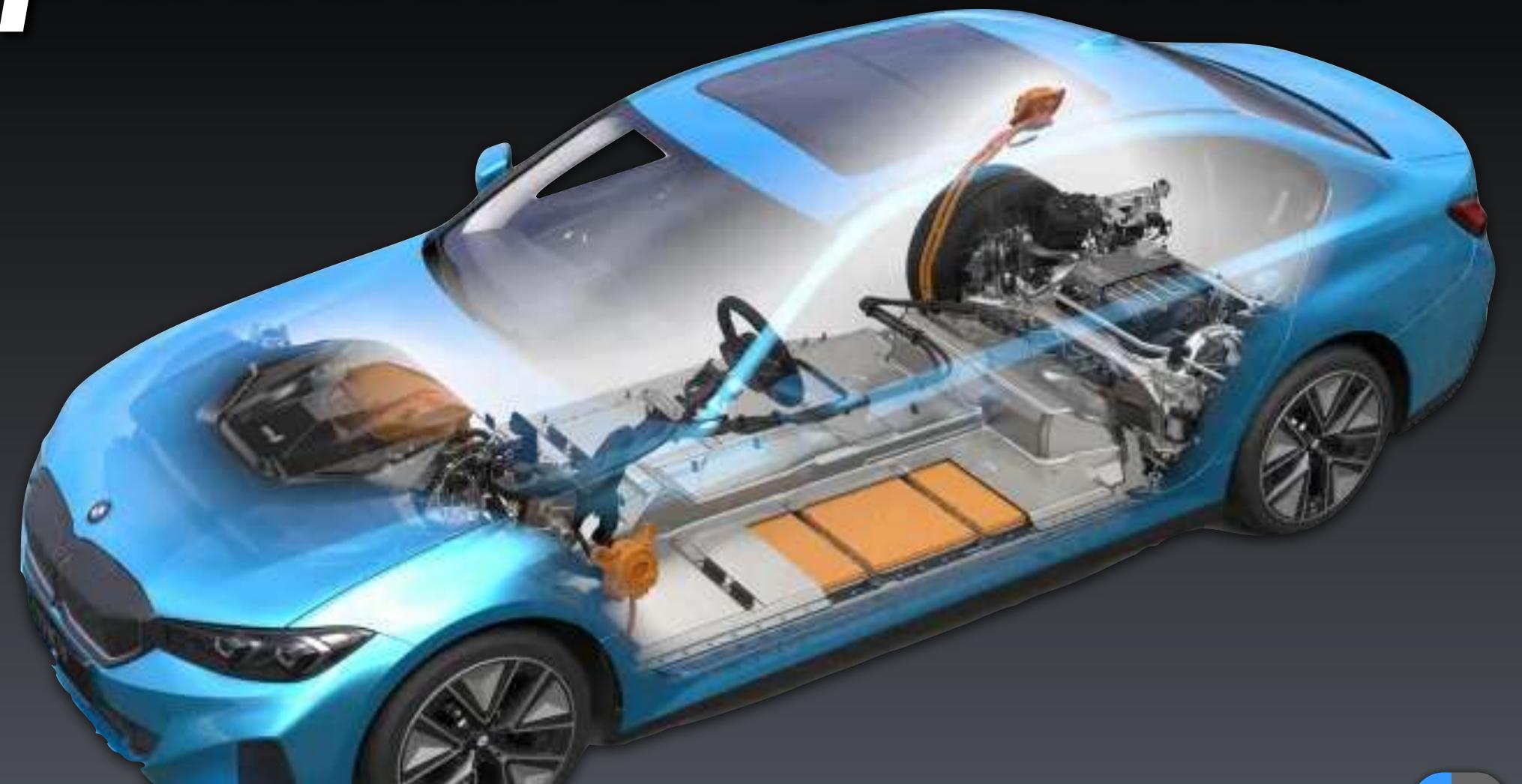








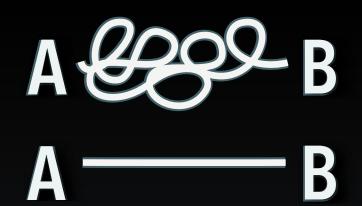




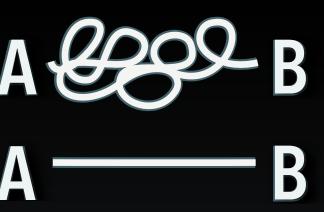




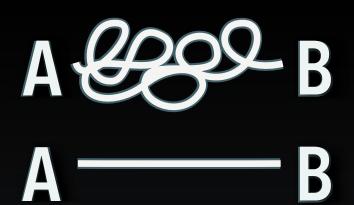








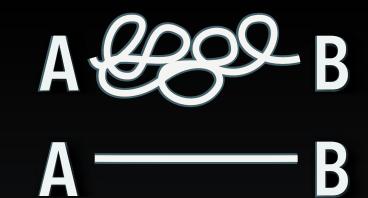




Clarity

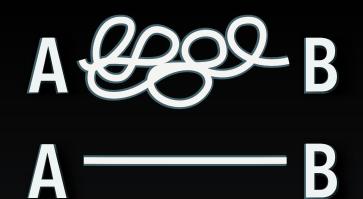


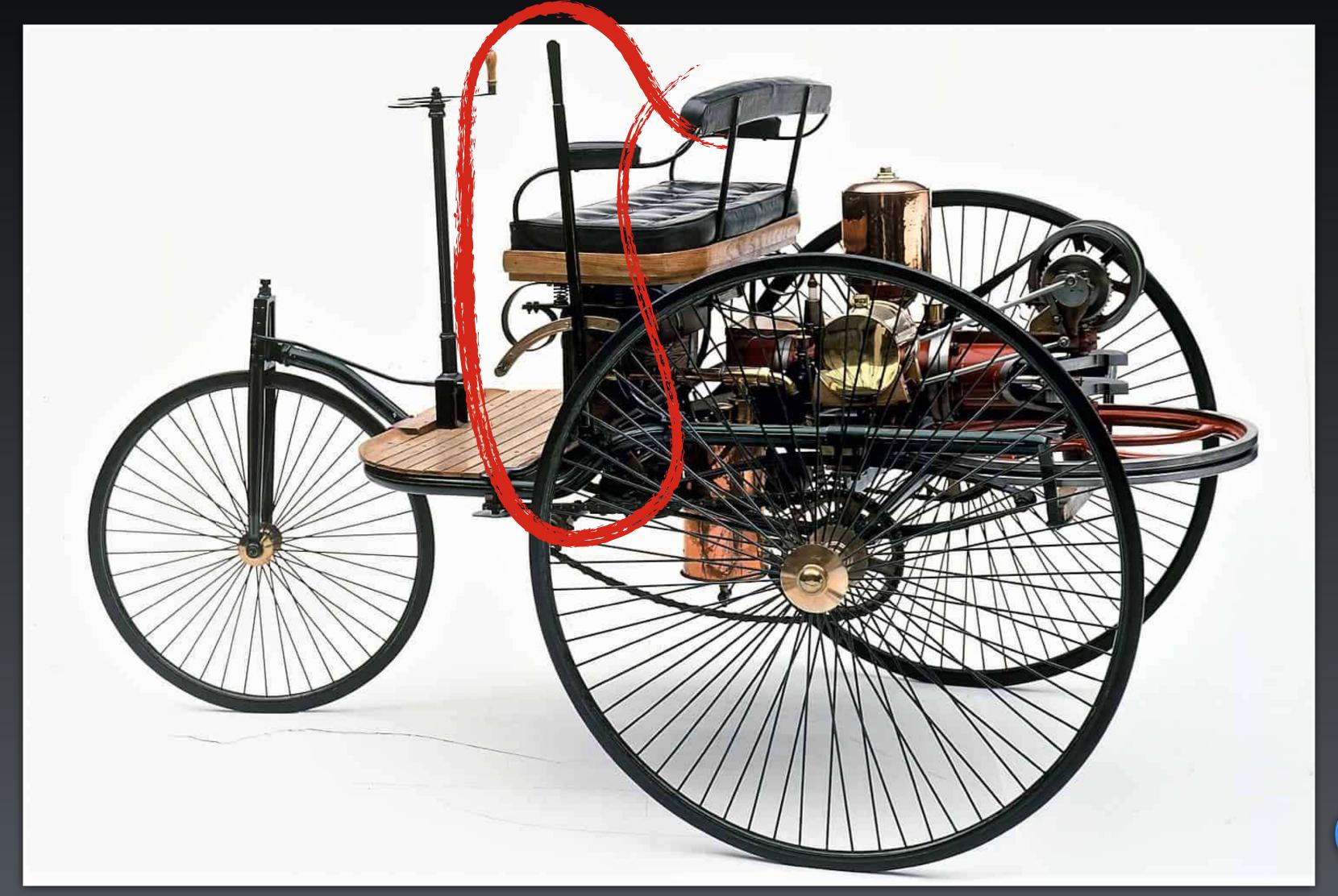




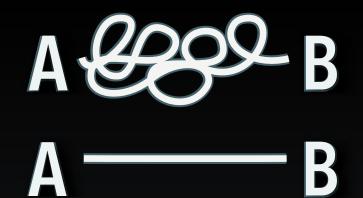


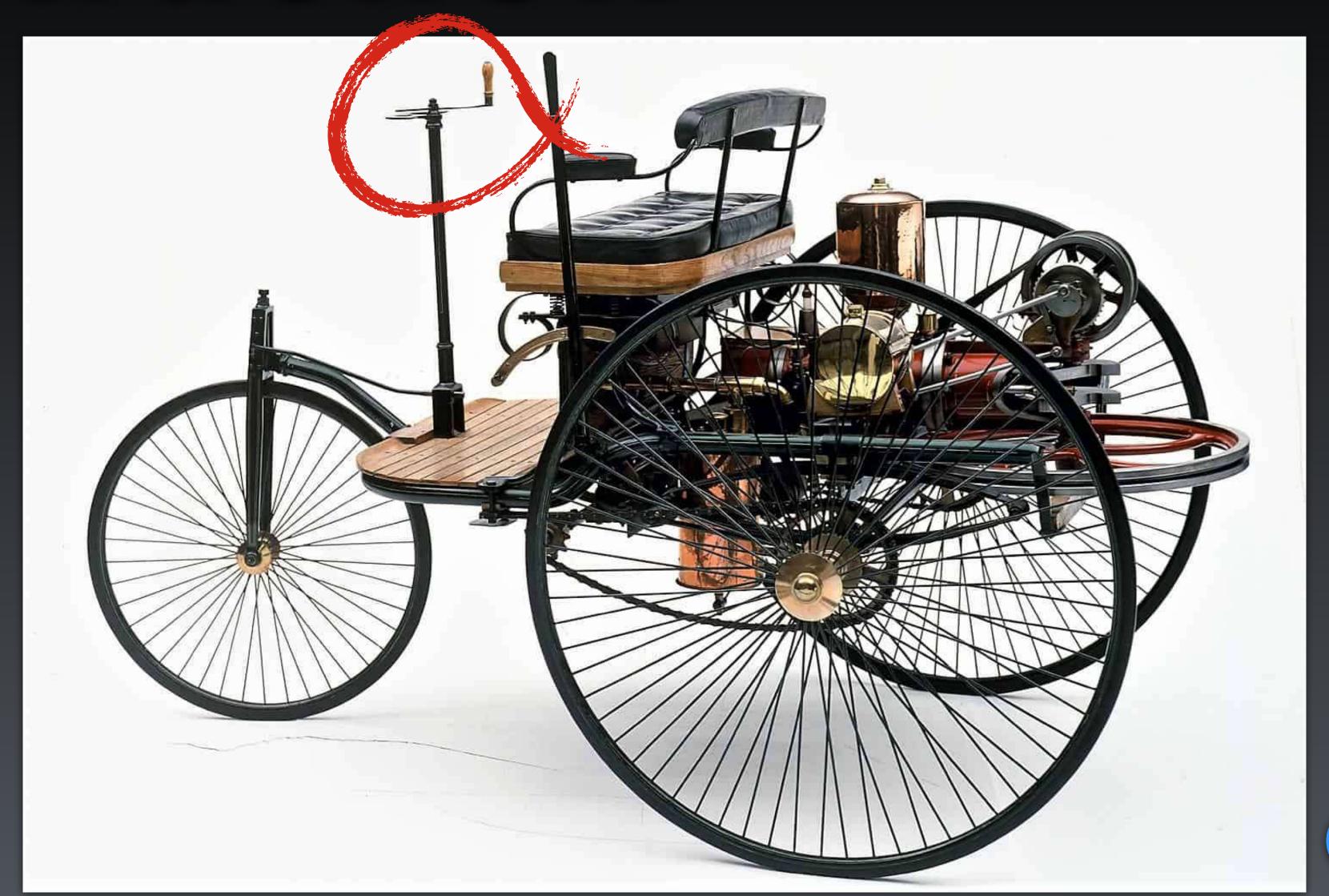
















A ----- E



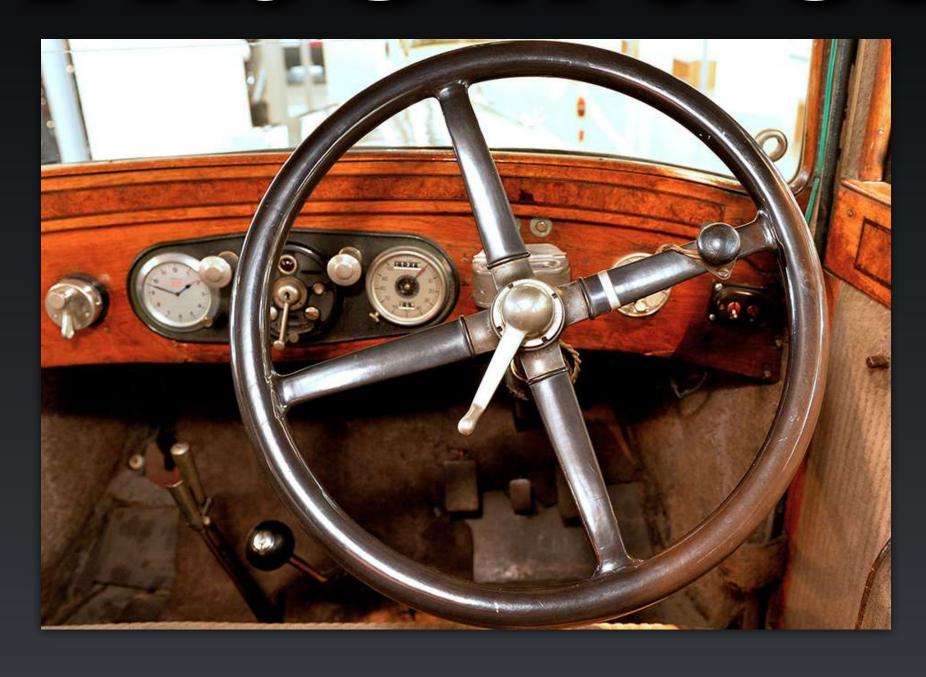


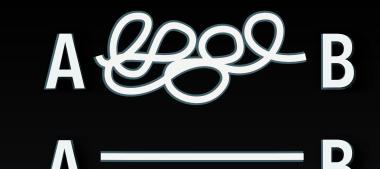


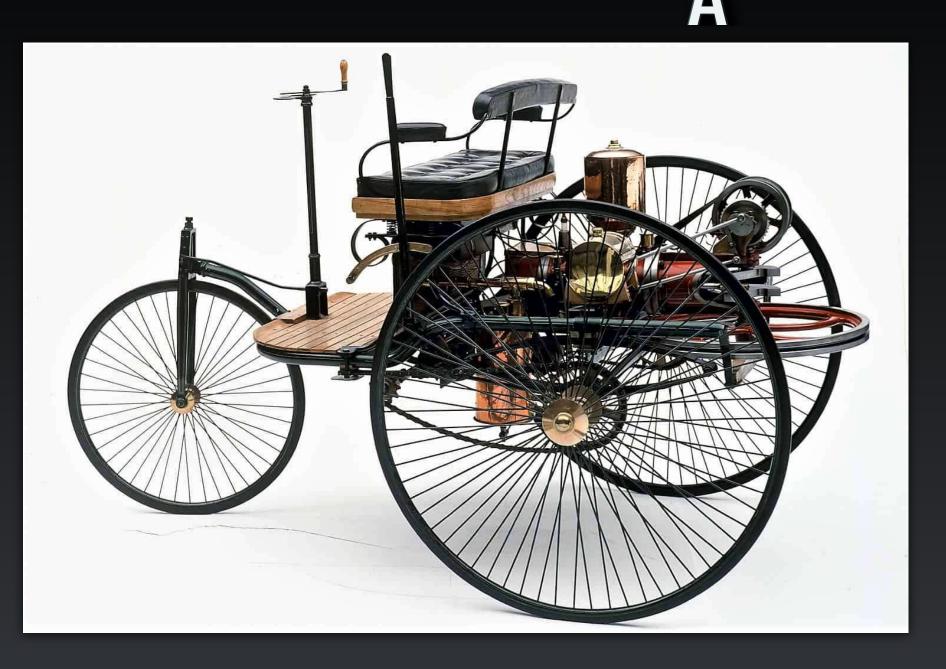
A ——— E







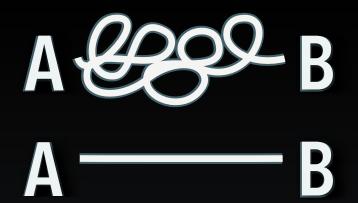






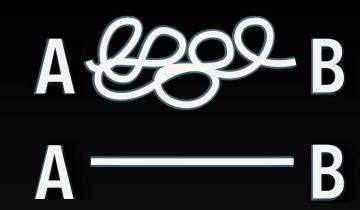


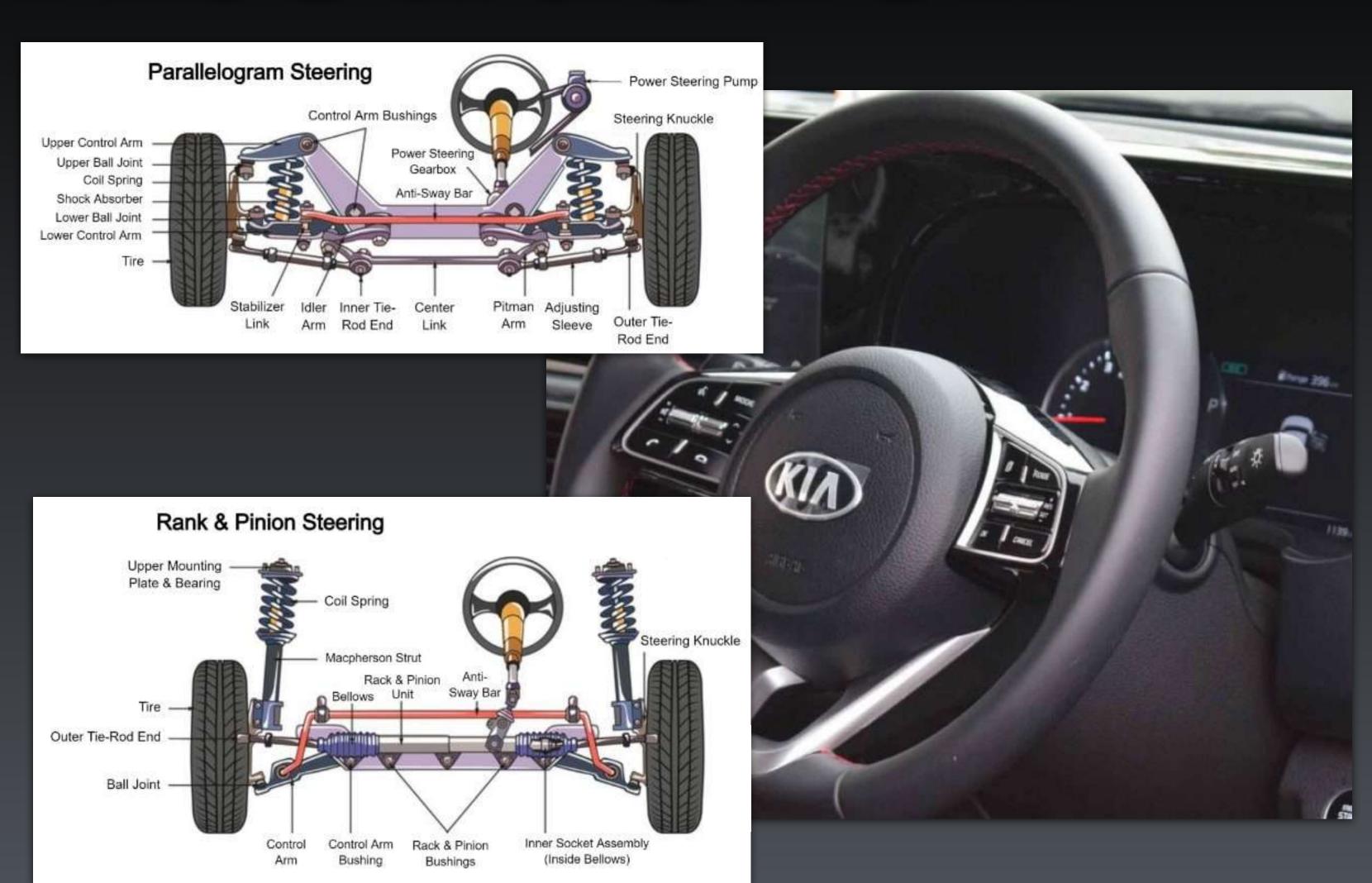




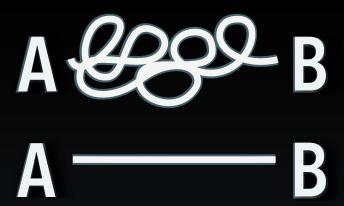


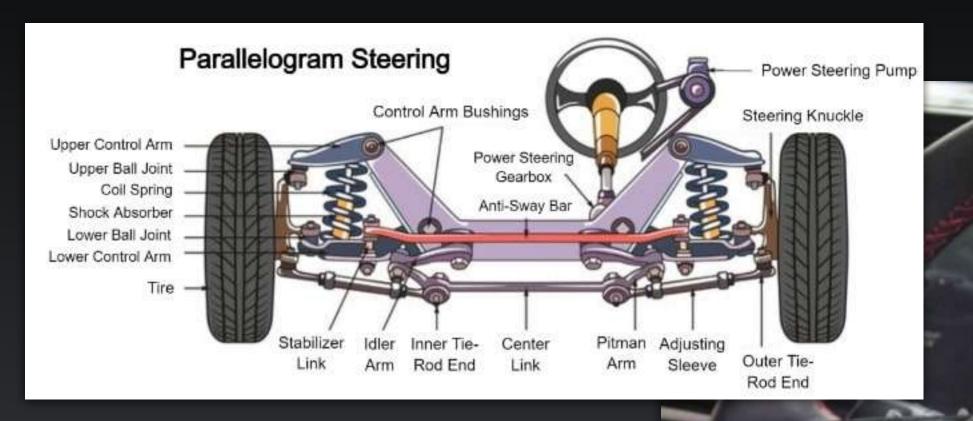


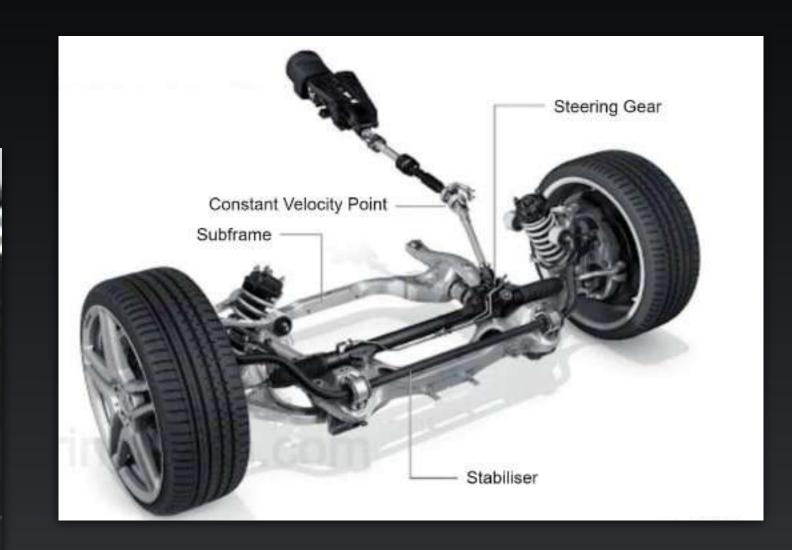


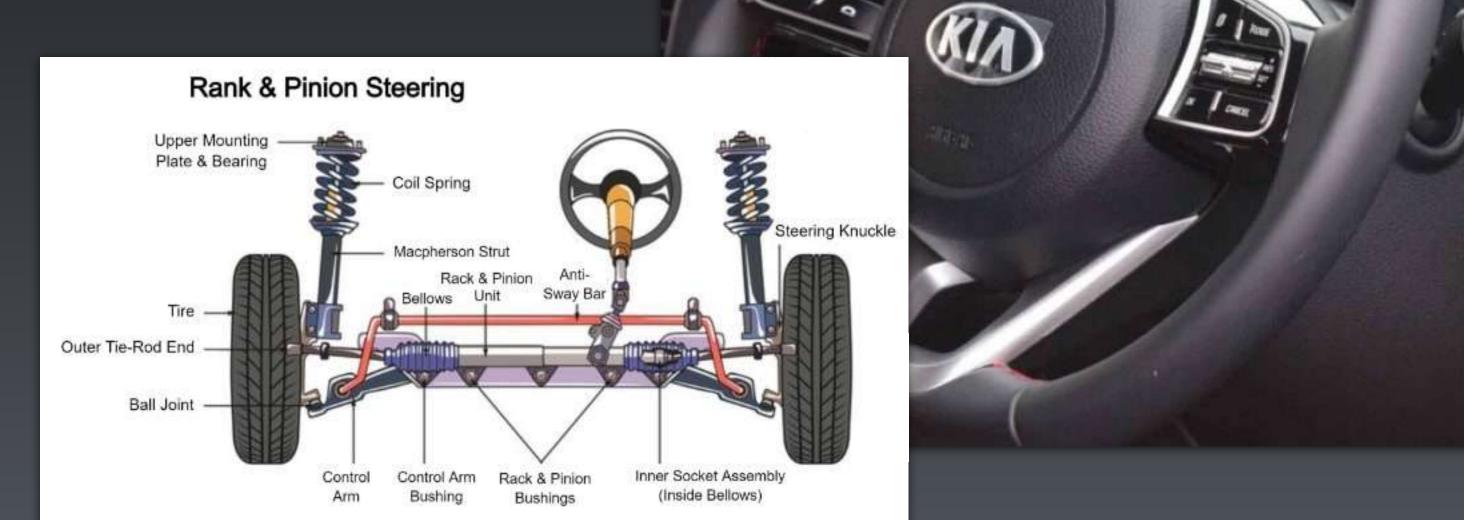


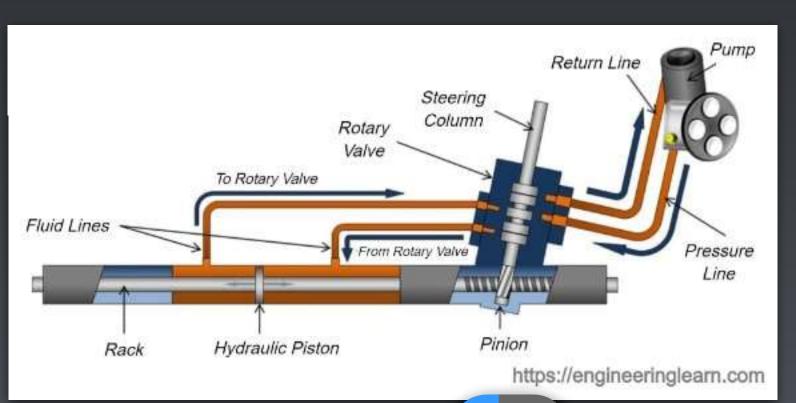




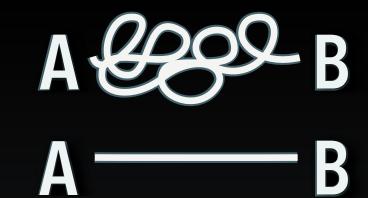


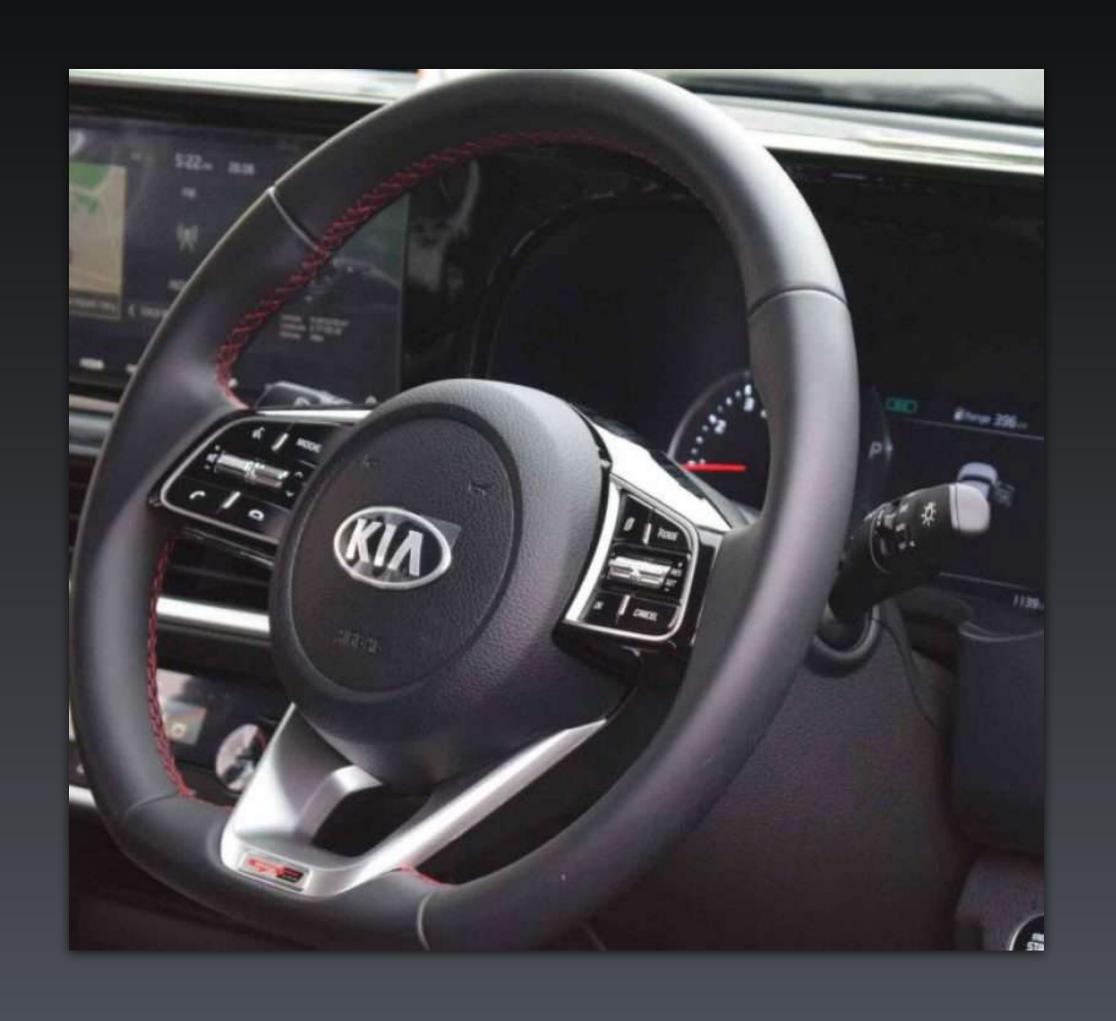




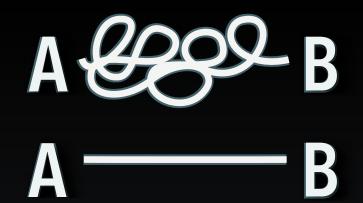




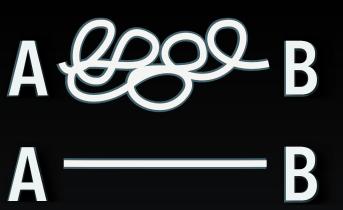




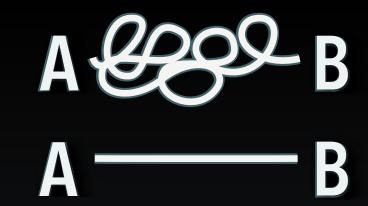


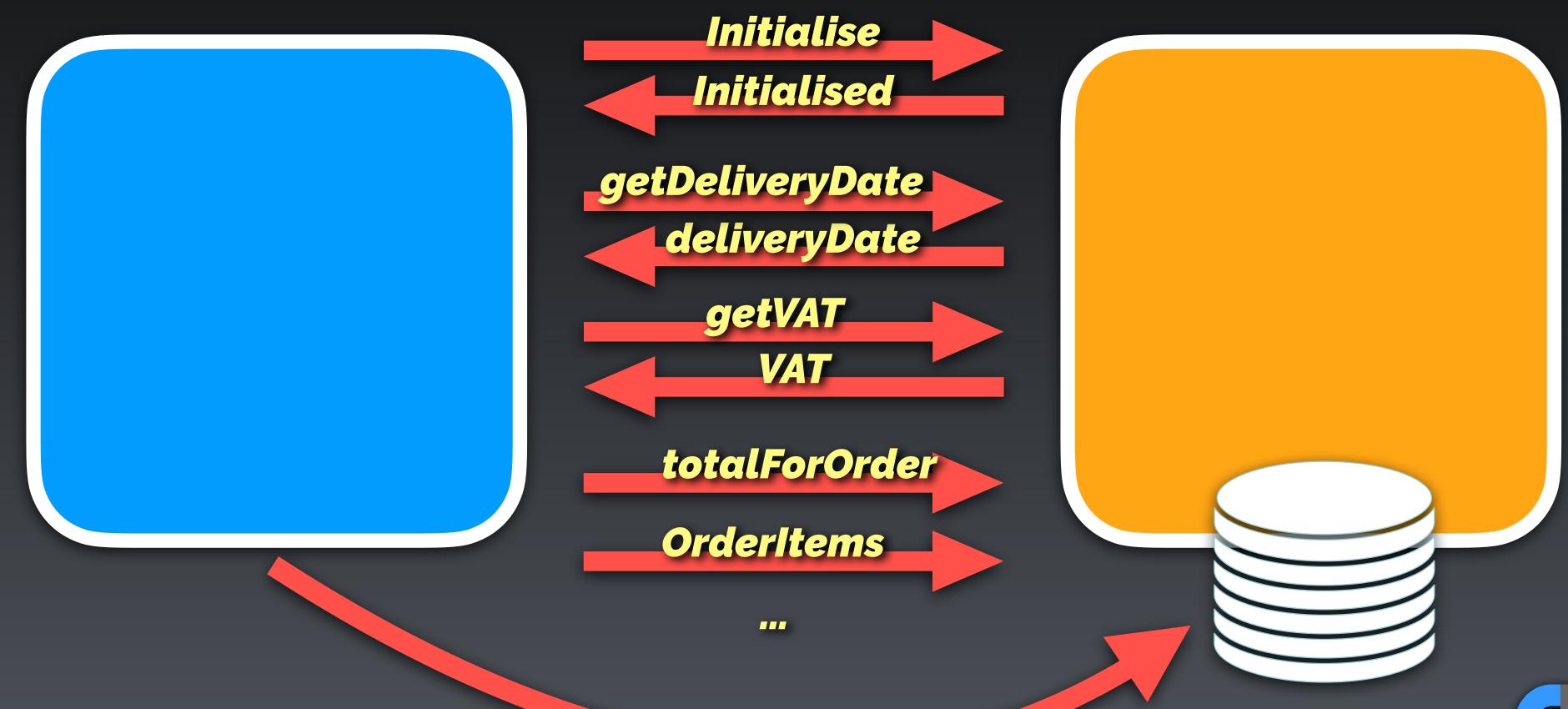






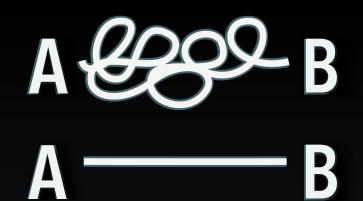






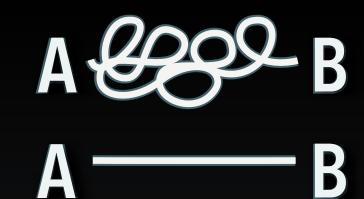
Confirm Order Placed





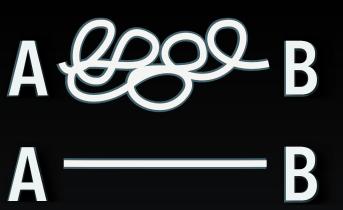




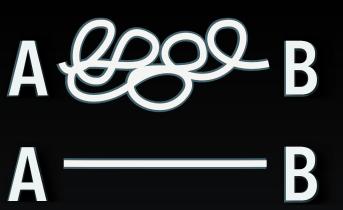




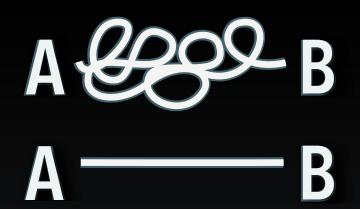










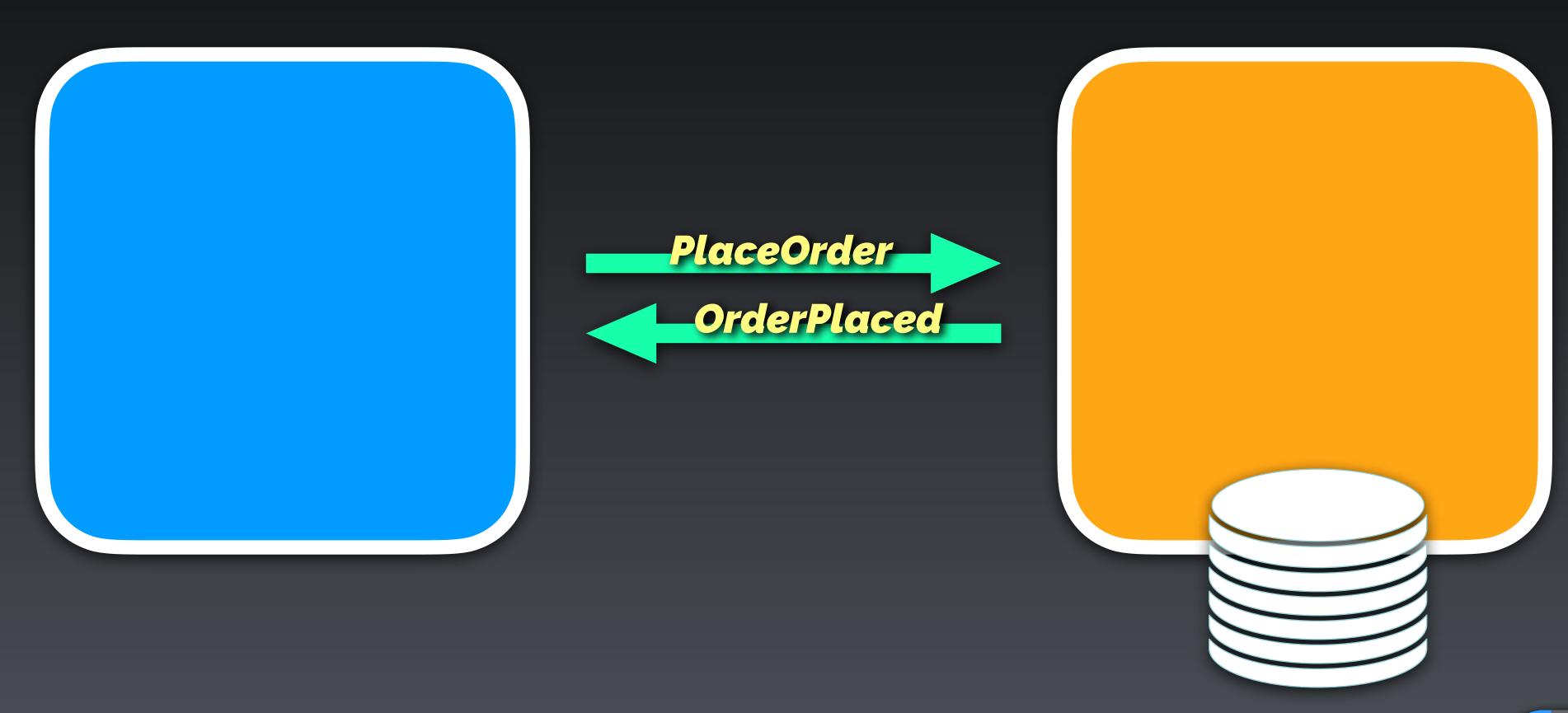














A LESS B

A R





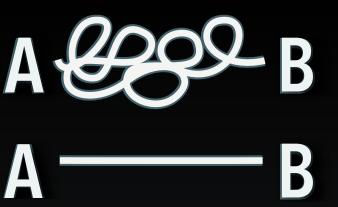
A LESS-B

Couping











A B A B

Coupling

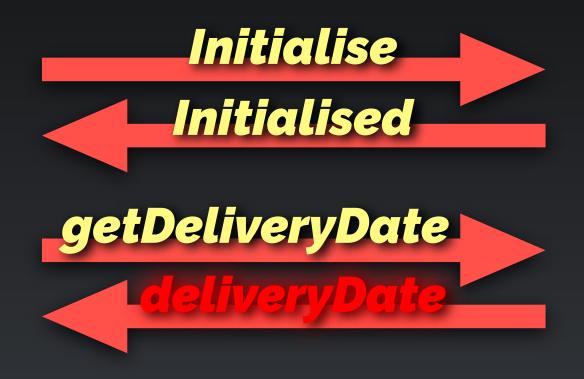


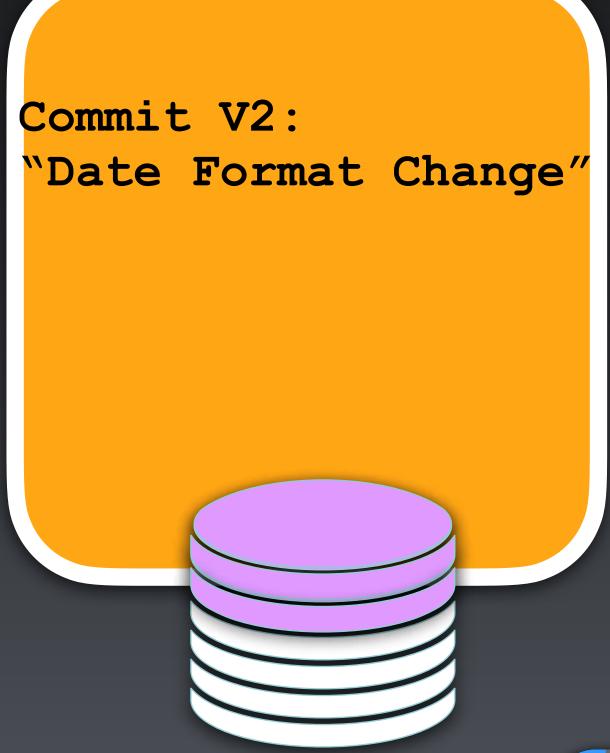
Commit V2: "Date Format Change"



A B B

Coupling









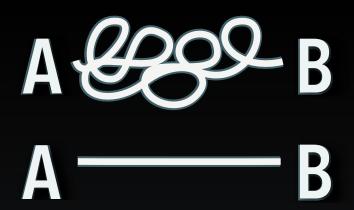
Coupling



Commit V2: "Date Format Change"

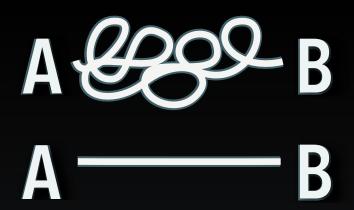


Coupling





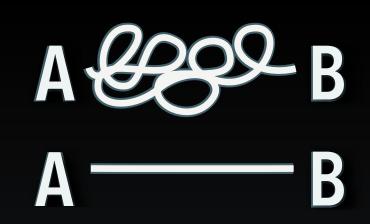
Coupling





Couping





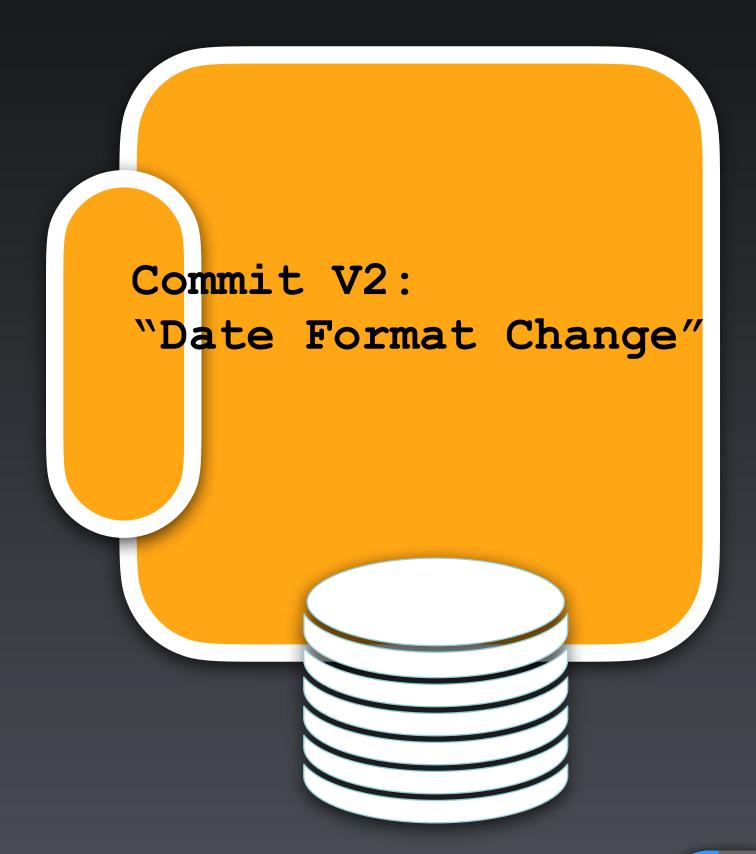




A B B

Coupling

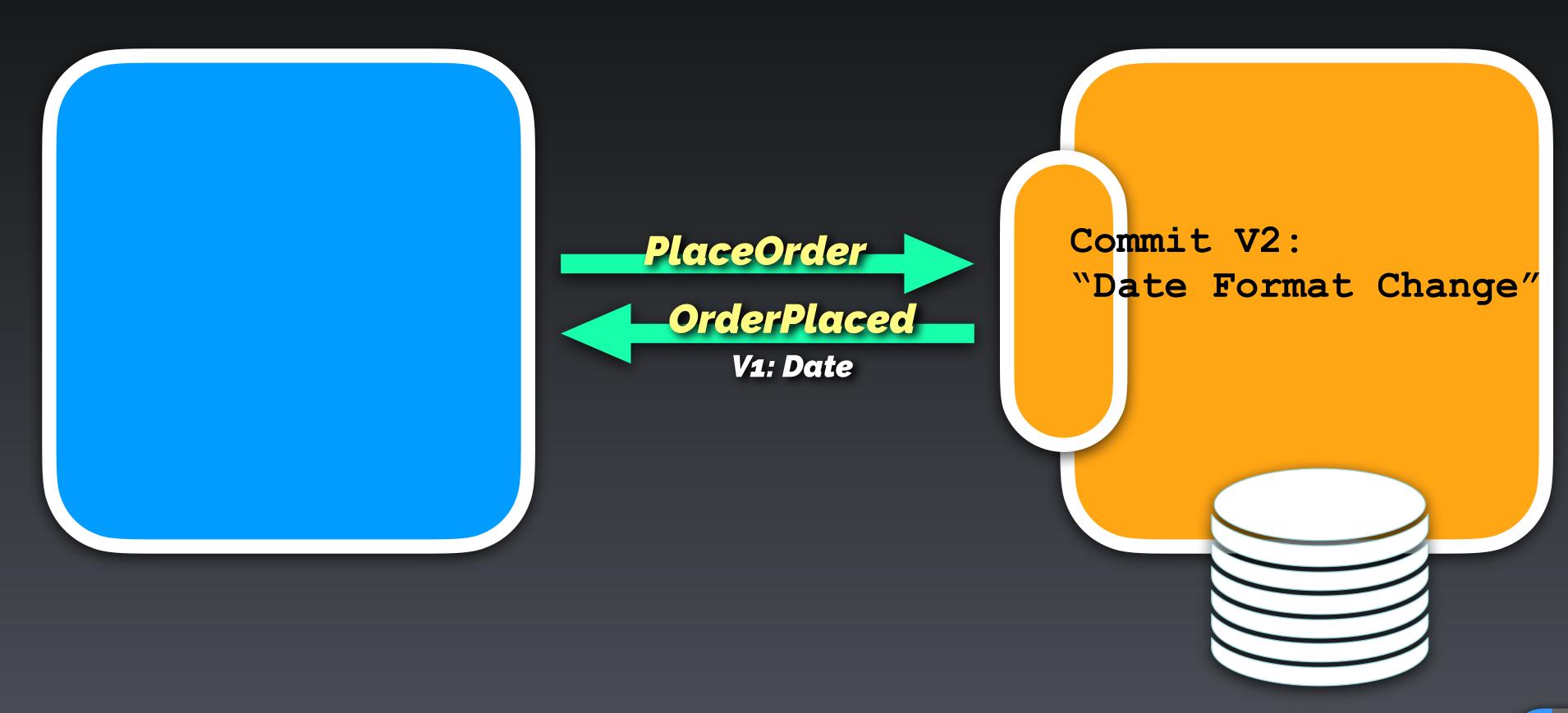






A B B

Couping





Applying the Guidelines



Managing Complexity



Managing Complexity

Modularity

Conesion

Separation of Concerns

Abstraction

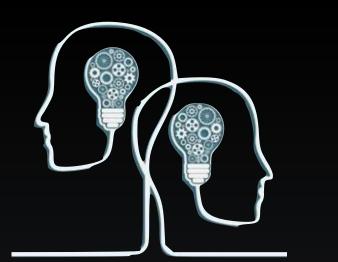
Couping



Managing Complexity







```
public class Car {
    private final Engine engine = new PetrolEngine();

public void start() {
    putIntoNeutral();
```

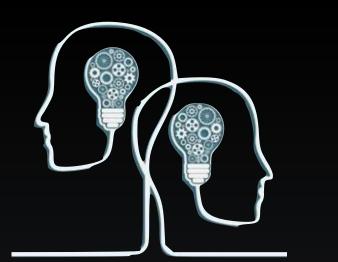
applyBrakes();

this.engine.start();

private void applyBrakes() {

private void putIntoNeutral() {

```
Continuous
Delivery Itd
```



```
public class Car {
    private final Engine engine = new PetrolEngine();

public void start() {
    putIntoNeutral();
```

applyBrakes();

this.engine.start();

private void applyBrakes() {

private void putIntoNeutral() {

```
Continuous
Delivery Itd
```

```
lass Car {
ivate final Engine engine = new PetrolEngine();
```

```
A B
```

```
public void start() {
    putIntoNeutral();
    applyBrakes();
    this.engine.start();
private void applyBrakes() {
private void putIntoNeutral() {
```

```
@Test
public void shouldStartCarEngine() {
    Car car = new Car();
    car.start();
    // Nothing to assert!!
```



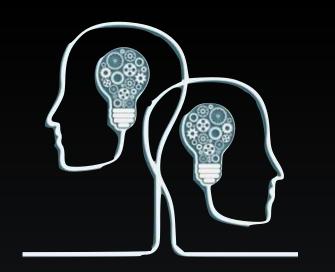
```
lass Car {
ivate final Engine engine = new PetrolEngine();
```

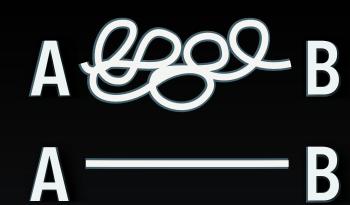
```
A B
A B
```

```
public void start() {
    putIntoNeutral();
   applyBrakes();
    this.engine.start();
private void applyBrakes() {
private void putIntoNeutral() {
```

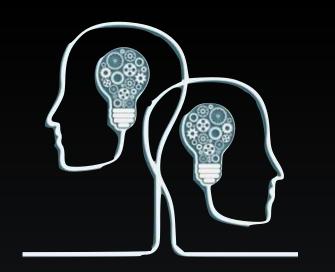
```
@Test
public void shouldStartCarEngine() {
    Car car = new Car();
    car.start();
    // Nothing to assert!!
```

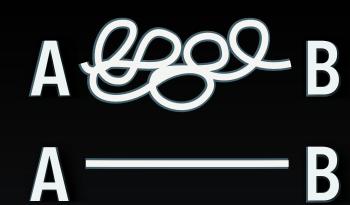




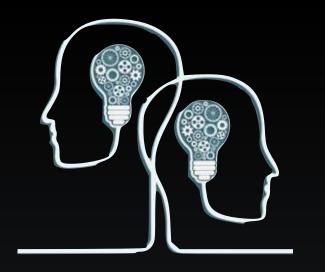






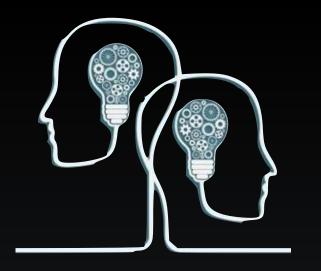






```
@Test
public void shouldStartBetterCarEngine() {
```

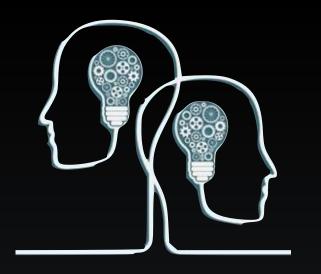




```
A B
```

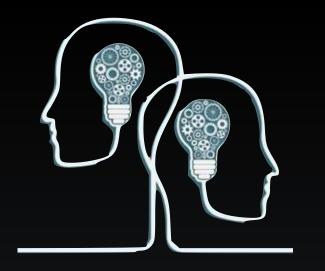
```
@Test
public void shouldStartBetterCarEngine() {
   assertTrue(engine.startedSuccessfully());
```





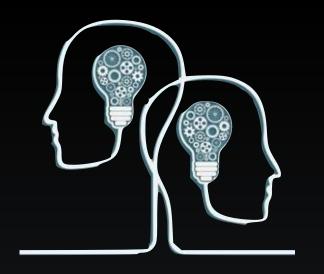
```
@Test
public void shouldStartBetterCarEngine() {
    car.start();
    assertTrue(engine.startedSuccessfully());
```





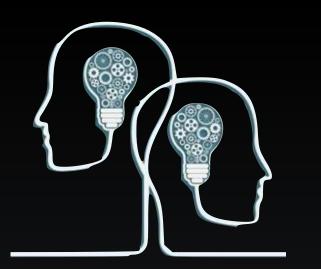
```
@Test
public void shouldStartBetterCarEngine() {
    BetterCar car = new BetterCar(engine);
    car.start();
   assertTrue(engine.startedSuccessfully());
```





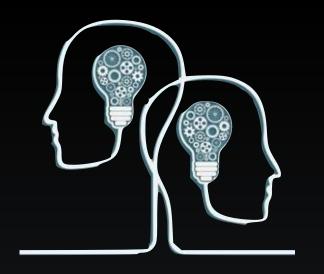
```
@Test
public void shouldStartBetterCarEngine() {
    FakeEngine engine = new FakeEngine();
    BetterCar car = new BetterCar(engine);
    car.start();
    assertTrue(engine.startedSuccessfully());
```





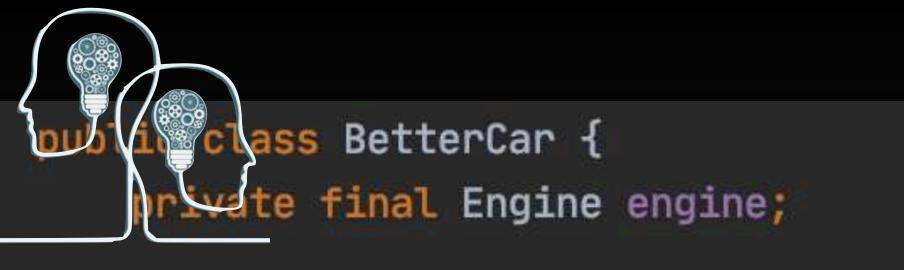
```
public class FakeEngine implements Engine {
    private boolean started = false;
    @Override
    public void start() {
        started = true;
    public boolean startedSuccessfully() {
        return started;
                                               y());
```





```
@Test
public void shouldStartBetterCarEngine() {
    FakeEngine engine = new FakeEngine();
    BetterCar car = new BetterCar(engine);
    car.start();
    assertTrue(engine.startedSuccessfully());
```



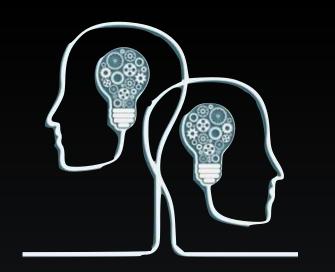


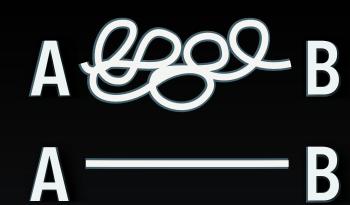
```
A B
```

```
public BetterCar(Engine engine) {
    this.engine = engine;
public void start() {
    putIntoNeutral();
    applyBrakes();
    this.engine.start();
private void applyBrakes() {
private void putIntoNeutral() {
```

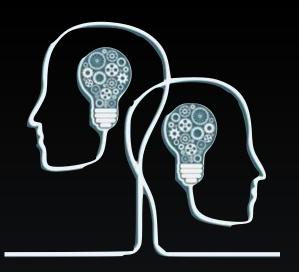
```
@Test
public void shouldStartBetterCarEngine() {
    FakeEngine engine = new FakeEngine();
    BetterCar car = new BetterCar(engine);
    car.start();
    assertTrue(engine.startedSuccessfully());
```



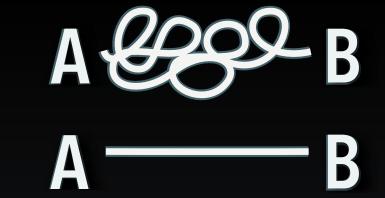




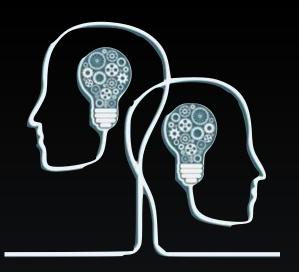




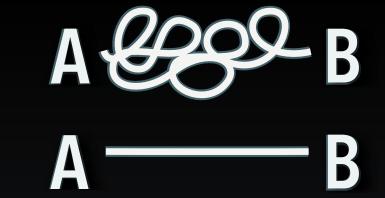
```
public class BetterCar {
    private final Engine engine;
    public BetterCar(Engine engine) {
        this.engine = engine;
    public void start() {
        putIntoNeutral();
        applyBrakes();
        this.engine.start();
    private void applyBrakes() {
    private void putIntoNeutral() {
```



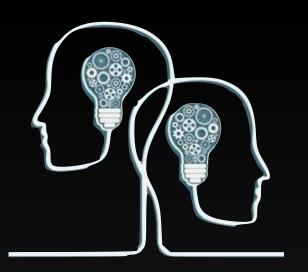




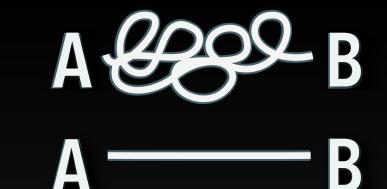
```
public class BetterCar {
    private final Engine engine;
    public BetterCar(Engine engine) {
        this.engine = engine;
    public void start() {
        putIntoNeutral();
        applyBrakes();
        this.engine.start();
    private void applyBrakes() {
    private void putIntoNeutral() {
```



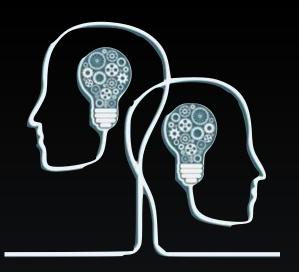




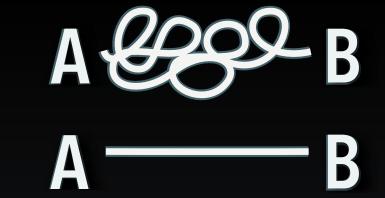
```
public class BetterCar {
           private final Engine engine;
           public BetterCar(Engine engine) {
              this.engine = engine;
              public class C
   private final Engine engine = new PetrolEngine();
```



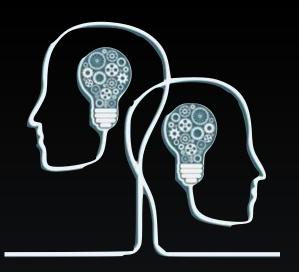




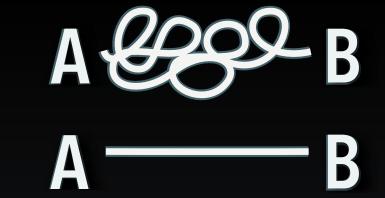
```
public class BetterCar {
    private final Engine engine;
    public BetterCar(Engine engine) {
        this.engine = engine;
    public void start() {
        putIntoNeutral();
        applyBrakes();
        this.engine.start();
    private void applyBrakes() {
    private void putIntoNeutral() {
```



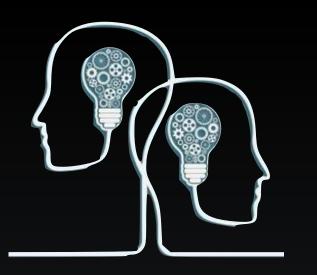




```
public class BetterCar {
    private final Engine engine;
    public BetterCar(Engine engine) {
        this.engine = engine;
    public void start() {
        putIntoNeutral();
        applyBrakes();
        this.engine.start();
    private void applyBrakes() {
    private void putIntoNeutral() {
```







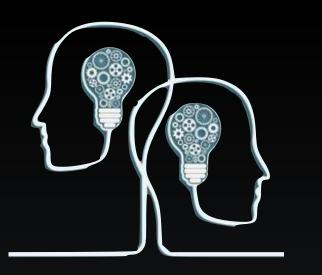
```
public class BetterCar {
    private final Engine engine;

public BetterCar(Engine engine) {
    this.engine = engine;
}
```

```
A B
```

```
public void createCars() {
```





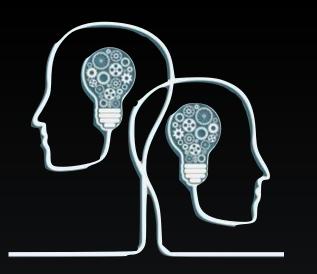
```
public class BetterCar {
    private final Engine engine;

public BetterCar(Engine engine) {
    this.engine = engine;
}
```

```
A B B
```

```
public void createCars() {
    BetterCar petrolCar = new BetterCar(new PetrolEngine());
```





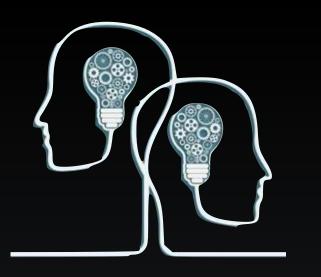
```
public class BetterCar {
    private final Engine engine;

public BetterCar(Engine engine) {
    this.engine = engine;
}
```

```
A B
```

```
public void createCars() {
   BetterCar petrolCar = new BetterCar(new PetrolEngine());
   BetterCar electricCar = new BetterCar(new ElectricEngine());
```





```
public class BetterCar {
    private final Engine engine;

public BetterCar(Engine engine) {
    this.engine = engine;
}
```

```
A B
```

```
public void createCars() {
   BetterCar petrolCar = new BetterCar(new PetrolEngine());
   BetterCar electricCar = new BetterCar(new ElectricEngine());
   BetterCar jetCar = new BetterCar(new JetEngine());
```





The Tools of Our Trade!

Optimise for Learning

Iteration Feedback Incremental Experimental Empirical

Optimise to Manage Complexity

Modularity Cohesion Separation of Concerns Abstraction Coupling

Continuous



· These Things Matter for 1 Reason



- · These Things Matter for 1 Reason
- · They Allow us to Change Our Software



Embrace Change!



Embrace Change!

JS Frameworks Change All the Time!

Requirements Change

Businesses Change

Technologies Change
People Change

Markets Change

Customers Change

Plans Change
Our Designs Change

Our Understanding Changes





The Quality of a System is Defined by Our Ability to Change it!









http://www.continuous-delivery.co.uk

Dave Farley



https://www.davefarley.net



@davefarley77



https://bit.ly/CDonYT

