

EMS31 Kwartaal 3 Week 2: Unittesten en Git submodules

Leerdoelen week 2 les 2. Je leert hoe je:

- C-code systematisch kan testen met behulp van het eenvoudige test-framework **GoogleTest**;
- een ontwikkelmethode voor software genaamd Test Driven Development (**TDD**) kan toepassen;
- code die je in meerdere applicaties wilt gebruiken in een **Git submodule** kan plaatsen;
- zo'n submodule kunt gebruiken onder Windows Subsystem for Linux (**WSL**);
- zo'n submodule kunt gebruiken in Code Composer Studio (**CCS**).

Waarom testen?

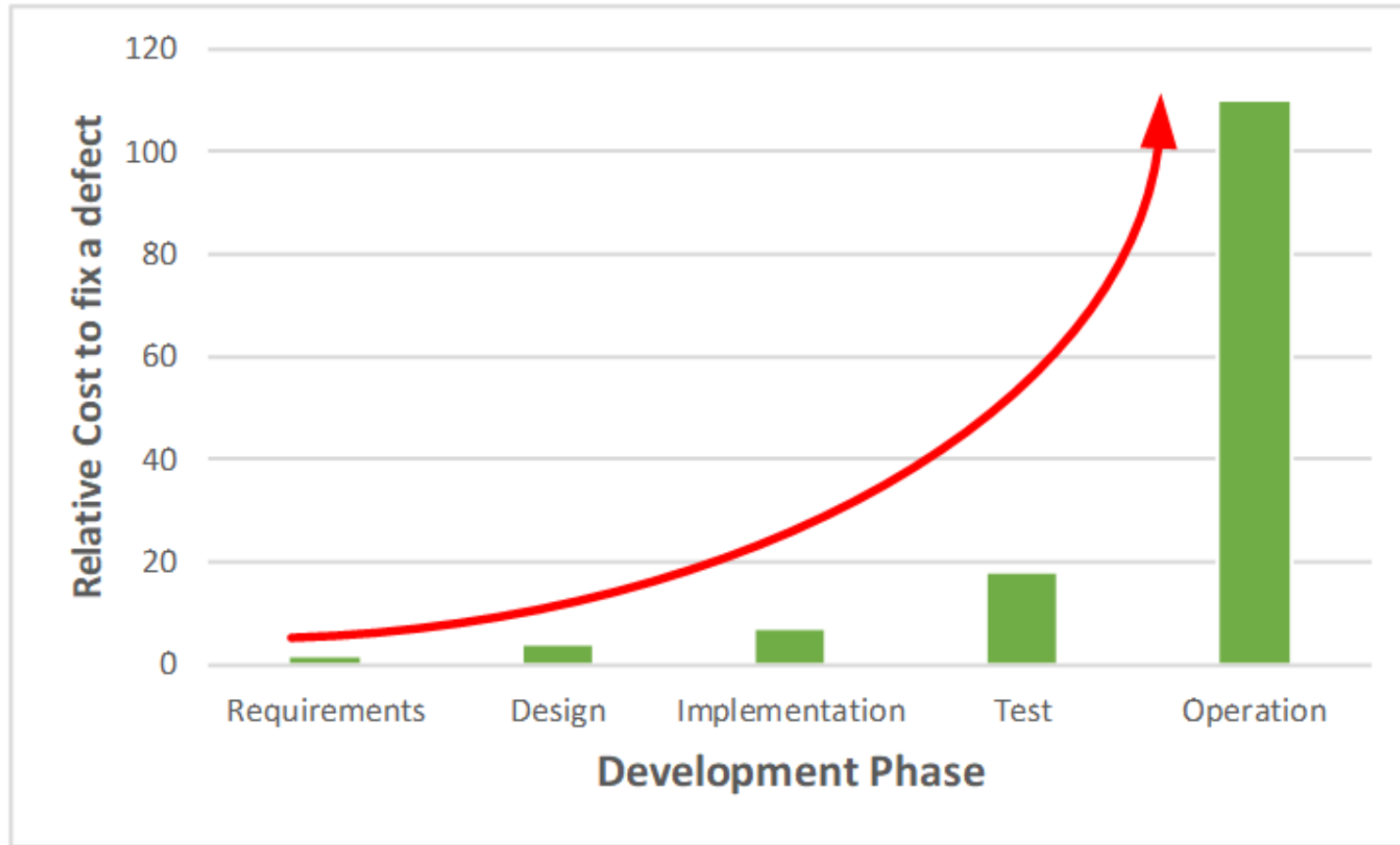
Foutvrije software ontwikkelen is erg moeilijk misschien zelfs onmogelijk.

Spraakmakende fouten:

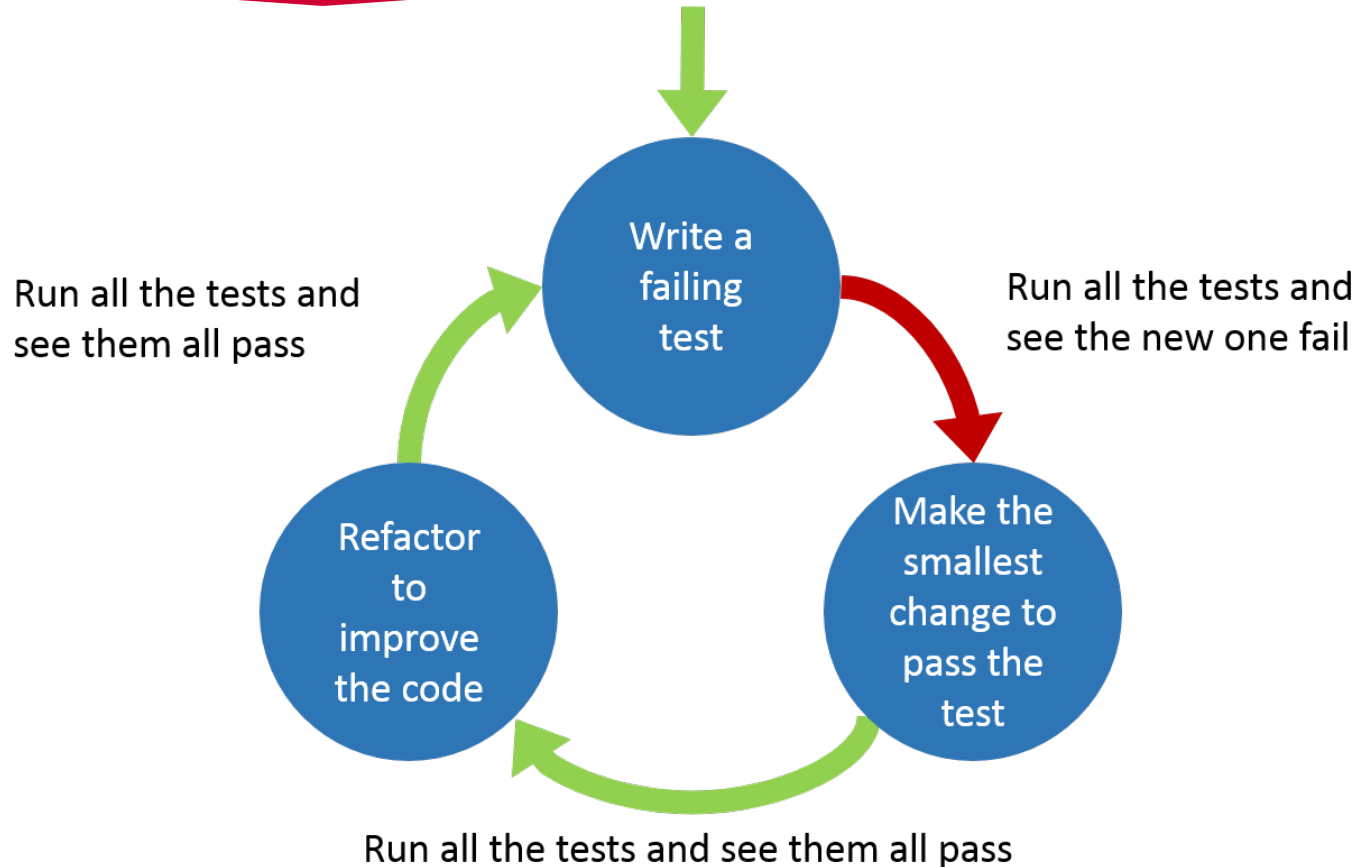
- 1996 – [Ariana 5 raket](#)
- 2013 – [Toyota versnelt als op de rem wordt geduwd](#)
- 2018 – [Boeing 737 MAX 8 stort neer](#)
- 2019 – [Metro Hoekse Lijn 2 jaar vertraagd en kost 175 million Euro extra](#)

Nog veel meer voorbeelden: <https://www5.in.tum.de/~huckle/bugse.html>
en https://en.wikipedia.org/wiki/List_of_software_bugs

Waarom unit's testen



Test Driven Development (TDD)

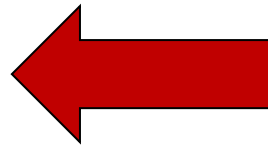


Bron plaatje: <https://hiddeninplainsight.co.uk/post/embedded-tdd/>

https://en.wikipedia.org/wiki/Test-driven_development

Er zijn veel test frameworks beschikbaar die het schrijven en uitvoeren van unittests vereenvoudigen:

- AceUnit
- Criterion
- CUnit
- Unity
- Catch2
- GoogleTest



<http://google.github.io/googletest/>

Zie: https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

Voorbeeld test module breuk

test_breuk.cpp

```
#include <gtest/gtest.h>
// Include C code in C++ bestand
extern "C"
{
    #include "breuk.h"
}

TEST(breuken, optellen)
{
    Breuk a = {1, 3}, b = {2, 4};

    // Als init niet werkt heeft verder testen geen zin
    ASSERT_EQ(a.teller, 1);
    ASSERT_EQ(a.noemer, 3);
    ASSERT_EQ(b.teller, 2);
    ASSERT_EQ(b.noemer, 4);

    Breuk c = add(a, b);
    // Check resultaat
    EXPECT_EQ(c.teller, 5);
    EXPECT_EQ(c.noemer, 6);
}
```

Voorbeeld test module breuk

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.27)
project(breuk_test)
# test_breuk moet een C++ programma zijn
add_executable(test_breuk test_breuk.cpp breuk.c)
set(CMAKE_C_FLAGS "${CMAKE_C_FLAGS} -std=c18 -Wall -Wextra -Wpedantic -g3 -O0")

target_link_libraries(test_breuk GTest::gtest_main)
include(CTest)
find_package(GTest REQUIRED)
include(GoogleTest)

gtest_discover_tests(test_breuk)
```

Zie: https://bitbucket.org/HR_ELEKTRO/ems31/raw/master/Programmas/breuk-test.zip

Voorbeeld test module breuk

```
[ems@LT-4SWKPV2 build]$ /home/ems/Opdrachten/breuk-test/build/test_breuk
Running main() from /usr/src/debug/gtest/googletest-1.14.0/googletest/src/gtest_main.cc
[=====] Running 3 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 3 tests from breuken
[ RUN      ] breuken.optellen
[          OK ] breuken.optellen (0 ms)
[ RUN      ] breuken.vermenigvuldigen
/home/ems/Opdrachten/breuk-test/test_breuk.cpp:30: Failure
Expected equality of these values:
  c.teller
    Which is: 2
  1

/home/ems/Opdrachten/breuk-test/test_breuk.cpp:31: Failure
Expected equality of these values:
  c.noemer
    Which is: 12
  6

...
2 FAILED TESTS
```

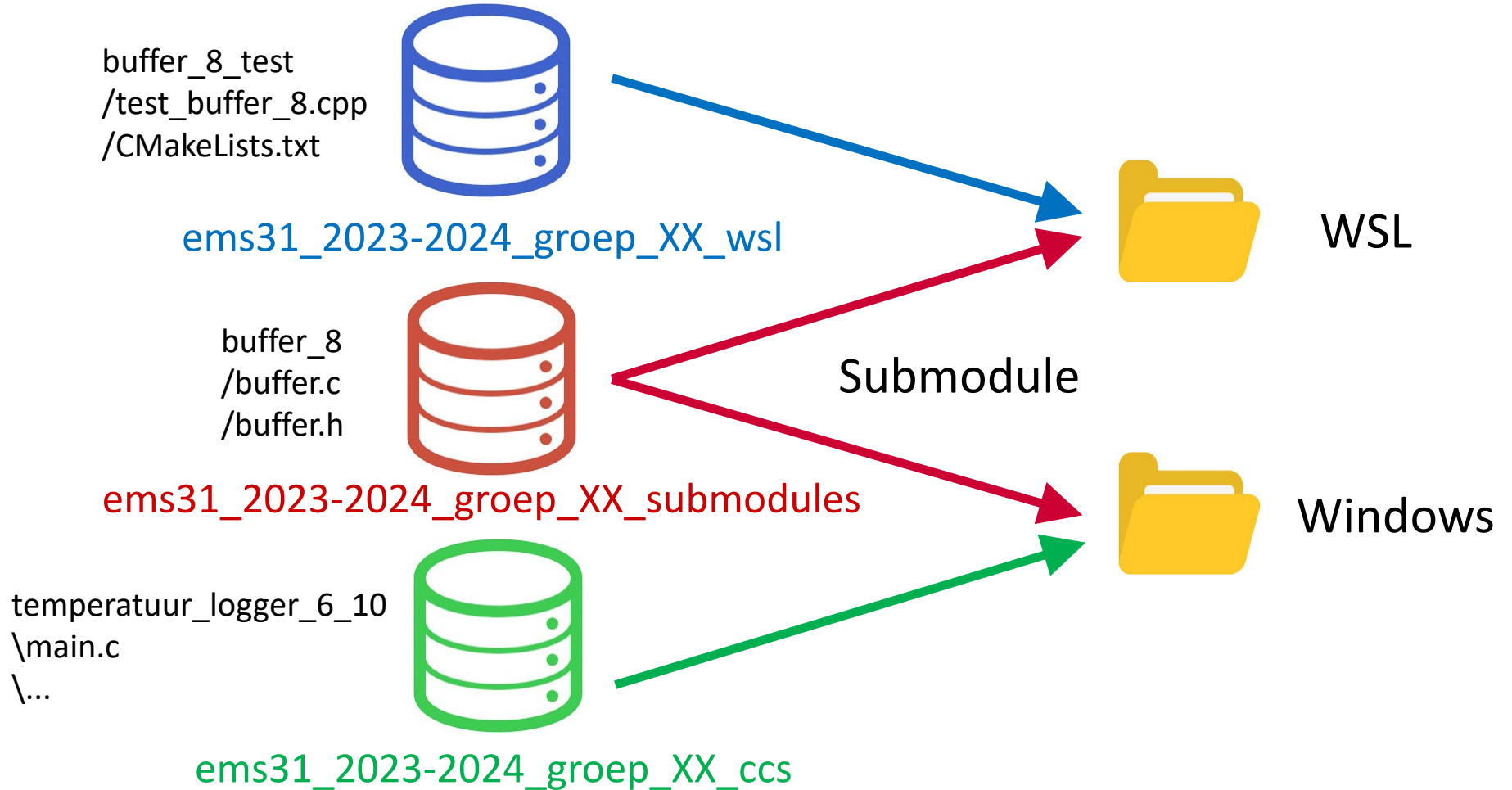
VS Code plugin C++ TestMate

The screenshot shows the Visual Studio Code interface with the TestMate plugin. On the left, the 'TESTING' sidebar displays a tree view of test results. The 'vermenigvuldigen' test is highlighted in red, indicating a failure. The main editor shows the C++ source code for 'test_breuk.cpp', which includes a test function 'TEST(breuken, vermenigvuldigen)'. The code defines two arrays 'a' and 'b', multiplies them, and checks the result. The output shows two failures: 'EXPECTED equality of these values: c.teller which is: 2 · 1' and 'EXPECTED equality of these values: c.noemer which is: 12 · 6'.

```
TEST(breuken, vermenigvuldigen)
{
    Breuk a = {1, 3}, b = {2, 4};
    Breuk c = mul(a, b);
    // Check resultaat
    EXPECT_EQ(c.teller, 1); # Failure: Expected
equality of these values: c.teller which is: 2 · 1
    EXPECT_EQ(c.noemer, 6); # Failure: Expected
equality of these values: c.noemer which is: 12 · 6
}
```

Zie: live demo

Git Submodules



Git Submodules

HR_ELE_STUD / EMS31_2023_2024

ems31_2023-2024_groep_00_wsl

Invite Clone ...

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

main Files Filter files

/

Name	Size	Last commit	Message
buffer_8_test		2024-01-14	buffer_8_test toegevoegd
ems31_2023-2024_groep_00_submodules [f2002bc45941]	40 B		
.gitmodules	171 B	2024-01-13	submodule toegevoegd
README.md	114 B	2024-01-13	README.md created online with Bitbucket

<https://www.atlassian.com/git/tutorials/git-submodule>

Is het (nog) zinvol om te leren programmeren?

EMBEDDED SYSTEMS



Bron: <https://algo.edu.vn/en/general/should-we-learn-coding-when-chatgpt-can-write-code/>

Is het zinvol om te leren programmeren?

Ja! Want:

- als je zelf niet kunt programmeren kun je door ChatGPT of andere AI tools gegenereerde code **niet begrijpen**, **beoordelen** en **aanpassen**;
- ChatGPT is getraind met *open source* code van ongelijke kwaliteit (veel code van embedded systemen is *close source*). De **kwaliteit** van de code is daarom vaak **middelmatig**.
- *“The latest versions of ChatGPT, GitHub Copilot, and Amazon CodeWhisperer generate correct code 65.2%, 46.3%, and 31.1% of the time”* Bron: [Evaluating the Code Quality of AI-Assisted Code](#) 22-10-2023.



- ChatGPT kan (bijna) alle **weekopdrachten** uit EMS31 zonder problemen uitvoeren.
- ChatGPT heeft moeite met de eindopdrachten.
- We hebben het getest bij EMS10:
Zie: https://bitbucket.org/HR_ELEKTRO/ems10/wiki/ChatGPT.md



- Het doel van de weekopdrachten van EMS31 is **niet** om de opdrachten op te lossen (dat kan ChatGPT ook).
- Het doel is **wel** om te leren hoe je een betere C-programmeur wordt.
 - Dit ter **voorbereiding** op je latere werk (en de eindopdrachten).

Moet je ChatGPT code laten genereren?

Nee! Want:

- dan leer je niet om het zelf te doen;
- ChatGTP gebruikt (vaak) constructies die (nog) niet behandeld zijn in de voorgaande lessen.



Als je eenmaal zelf goed kunt programmeren kan je ChatGTP wel gebruiken als **assistent** die je help om bepaalde ‘standaard’ stukjes code te genereren.

Kan ChatGPT je helpen?

Je kunt, als je wilt, ChatGPT **wel** gebruiken om je te **helpen**.

- Zo kun je ChatGPT bijvoorbeeld vragen om **stap voor stap uit te leggen** hoe een stukje C-code of C++-code werkt.
- Ook kan ChatGPT helpen bij het **testen, debuggen** of **optimaliseren** van een programma.



Kan ChatGPT je helpen?



You

Schrijf unittests om de volgende C module te testen:

```
// put value i in buffer if buffer is not full
```

```
// returns true on success or false otherways
```

```
extern bool buffer_put(int i);
```

```
// get value from buffer and writes it to *p if buffer not empty
```

```
// returns true on success or false otherways
```

```
extern bool buffer_get(int *p);
```

```
// returns true when buffer is full or false otherways
```

```
extern bool buffer_is_full(void);
```

```
// returns true when buffer is empty or false otherways
```

```
extern bool buffer_is_empty(void);
```

Maak gebruik van GoogleTest




Kan ChatGPT je helpen?

ChatGPT

Hier is een voorbeeld van hoe je GoogleTest kunt gebruiken om de bovenstaande C-module te testen:



c

 Copy code

```
#include <gtest/gtest.h>
#include "buffer.h" // inclusie van de header die de functies declareert

// Test voor buffer_put
TEST(BufferTest, PutTest) {
    EXPECT_TRUE(buffer_put(5)); // voeg een waarde toe aan de buffer
}
```

ChatGPT geeft **één** simpele test voor elke functie.
ChatGPT snapt **niet** dat de module in C geschreven is.

Kan ChatGPT je helpen?

EMBEDDED SYSTEMS



You

Schrijf ook een test om het FIFO gedrag te testen



Kan ChatGPT je helpen?



ChatGPT

Hier is een test om het FIFO (First-In-First-Out) gedrag van de buffer te testen:



```
c Copy code
#include <gtest/gtest.h>
#include "buffer.h" // inclusie van de header die de functies declareert

// Test voor het FIFO-gedrag van de buffer
TEST(BufferTest, FIFOTest) {
    // Voeg enkele waarden toe aan de buffer
    buffer_put(1);
    buffer_put(2);
    buffer_put(3);

    // Haal waarden uit de buffer en controleer of ze in de juiste volgorde komen
    int value;
```

Kan slimmer met een for-loopje
Test niet of put geslaagd is.

Onze testcode

test_buffer_8.cpp

```
TEST(buffer, FIFO_behavior)
{
    // write 1, 2 and 3 into the buffer
    for (int i = 1; i <= 3; i++)
    {
        EXPECT_TRUE(buffer_put(i));
        EXPECT_FALSE(buffer_is_full());
        EXPECT_FALSE(buffer_is_empty());
    }
    // test if values can be retrieved in FIFO order from the buffer
    int retrieved_value;
    for (int i = 1; i <= 3; i++)
    {
        EXPECT_TRUE(buffer_get(&retrieved_value));
        EXPECT_EQ(retrieved_value, i);
        EXPECT_FALSE(buffer_is_full());
        if (i == 3)
        {
            EXPECT_TRUE(buffer_is_empty());
        }
        else
        {
            EXPECT_FALSE(buffer_is_empty());
        }
    }
}
```



Mag ChatGPT je helpen?

Bij EMS31 **mag** je ChatGPT of andere AI Coding Assistant (zoals [CoPilot](#)) gebruiken.



- Ook bij de **eindopdrachten**.
- Om plagiaat te voorkomen dien je dit in je verslag te **vermelden!**
- Zie: <https://hint.hr.nl/nl/HR/Over-de-HR/Kwaliteit-en-onderwijs/chatgpt/>
- Zie: <https://www.elegantthemes.com/blog/wordpress/best-ai-coding-assistant>

Belangrijk als je voor een bedrijf werkt:

ChatGPT

Tips for getting started

Ask away

ChatGPT can answer questions, help you learn, write code, brainstorm together, and much more.

Don't share sensitive info

Chat history may be reviewed or used to improve our services. Learn more about your choices in our [Help Center](#).

Check your facts

While we have safeguards, ChatGPT may give you inaccurate information. It's not intended to give advice.

Okay, let's go

Dynamisch geheugenallocatie



Aan de slag!

Aan de slag met [Opdrachten_Week_3.2.pdf](#)

