

# **Datorteknik**

## **Konstruera för felsökning**

Michael Josefsson

Version 0.4 2017-01-12

# Innehåll

<b>1. Konstruera för felsökning</b>	<b>3</b>
1.1. Om det inte fungerar? . . . . .	3
1.2. Feltyper . . . . .	4
1.3. Varför fel? . . . . .	4
1.3.1. Slumpfel . . . . .	5
1.4. Konstruera för felsökning . . . . .	6
1.4.1. Symptom. . . . .	7
1.4.2. Identifikation. . . . .	7
1.4.3. Åtgärda. . . . .	7
1.4.4. Förhindra. . . . .	7
<b>A. Citatsamling</b>	<b>13</b>

# 1. Konstruera för felsökning

Det är lika bra att vänja sig vid det: Inget fungerar som man tänkt sig. I varje fall inte direkt.

För att ändå lyckas med det man företar sig måste man använda återkoppling och successivt, genom upprepade små korrekitioner, komma framåt i rätt riktning.

För att kunna göra alla dessa små steg i rätt riktning måste vi redan från början konstruera så att alla apparatens beteenden betyder något. Det vill säga, vi måste *konstruera med en sådan struktur* att vi kan peka vad som inte fungerar när det inträffar. Vi måste redan från början *konstruera för att kunna felsöka!*

## 1.1. Om det inte fungerar?

Rubriken är sannolikt felaktig, det handlar inte *om* det inte fungerar eller inte, utan *när* det inte fungerar. Varje uppgift vi måste planera för är så komplex att vi kan räkna med att allt inte kommer att fungera helt perfekt från början.

Men även om rubriken ändrades till ”när det inte fungerar” vore den inte helt rättvisande. Den största frågan är ”hur **vet** vi att det inte fungerar”?

Ja, hur ska vi veta att något inte fungerar? För att behöva laga något måste vi först veta att det är trasigt. Erfarenheten säger oss att något är fel om en lampa inte tänds när vi vrider om strömbrytaren, eller om till exempel motorvarvet inte går upp när vi trycker ned gaspedalen.

Alltså: **Något fungerar inte när vi inte får ett väntat resultat!**

Vad är då ett förväntat resultat? Ofta är ett förväntat resultat det som förväntas minst. I första exemplet ovan är det förväntade att lampan ska tändas när vi vrider på strömbrytaren. Gör den inte det blir vi *förvånade*.

Förvånade blir vi när erfarenheten säger oss en sak och situationen en annan. I verkliga livet har vi en livserfarenhet som hjälper oss. Det behöver inte vara så alltid. I många konstruktioner måste vi lita på teorier och simuleringar som förutsäger vad vi ska förvänta oss.<sup>1</sup> Vi kanske dessutom inte lätt kan säga någonting

---

<sup>1</sup>För att vara bra felsökare måste vi alltså först vara väl insatta i dessa teorier och simuleringar med mera.

## 1. Konstruera för felsökning

om resultatet, utan att utföra mer eller mindre krångliga mätningar på konstruktionen. Hur vet vi om något fungerar i till exempel en mikroprocessor, om vi inte kan mäta på dess utsignaler? Det kan vi inte. Vi måste mäta!

### 1.2. Feltyper

Det kan uppstå fel av flera orsaker och alla fel är inte likadana. Enkla fel, dvs sådana som är lätta att inse att de är fel, är ofta påtagliga och kanske drastiska:

- Förstärkaren säger ”poff” och eldslågor slår ut ur den. Allmän erfarenhet säger att en förstärkare ska förstärka signaler — inte låta ”poff”. Och definitivt inte börja brinna,
- Någonting i bilen låter ”kladonk-kladonk” när vi svänger. Det är inte som det brukar.<sup>2</sup>

Alla avvikelser behöver dock inte vara så påtagliga:

- Spänningsaggregatets switchtransistor är mycket varm.<sup>3</sup>

För att kunna avgöra vad som är ”som det brukar” och inte måste vi ha erfarenhet om just denna situation. Det är inte säkert vi har den erfarenheten. Då måste vi återigen mäta! Det kan också handla om avvikelser som man inte kan se innan man mätt på systemet med oscilloskop, logikanalysator eller universalinstrumet eller vad man nu har för utrustning:

- Processorklockan är 4 MHz och inte 3.58 MHz som vi tror.
- Kollektorströmmen är 540 mA istället för väntade 210 mA.
- Kurvformen är distorderad vid större positiva utslag.

### 1.3. Varför fel?

När väl avvikelserna är identifierade måste man ställa sig nästa fråga: Vad är det som gör att det blev fel? Hur kan avvikelserna uppstå?

Ofta finns inte enkla svar på denna fråga. Inte sällan är avvikelserna ett resultat av en händelsekedja:

---

<sup>2</sup>Dags att byta drivknutar brukar det tyda på.

<sup>3</sup>Här kan man fundera på var gränsen mellan *varm* och *mycket varm* och *farligt varm* går!

## 1. Konstruera för felsökning

- Förstärkaren sade ”poff” för att den kördes mot en felaktig högtalare. Är detta orsaken? Kanske, men det är nog fortfarande ett symptom och ingen orsak eftersom en välkonstruerad förstärkare har skyddskretsar som ser till att den inte förstörs av en felaktig last.

Orsaken *kan* vara att skyddskretsarna satts ur spel *eftersom* de tappat matningsspänning *eftersom* en kallödning uppstått på kretskortet *eftersom* detta blivit uppvärmt av transformatorn *eftersom* den är satt på 220 volt växelspanning fast vi i vägguttaget har 240 volt.

- Spänningsaggregates transistorer kan vara varma för att kylflänsen är otillräcklig men *kanske* är det istället kylfläkten som inte fungerar? *Kanske* värmeledningspastan släppt? *Kanske* är det...

Det finns som du märker en hel uppsjö av möjliga anledningar till att felet uppstått. Vid felsökning är fantasi en stor tillgång. Som konstruktör gäller det att känna till **så mycket som möjligt** för att kunna felsöka så bra som möjligt. Om du bara reflexmässigt byter till, kanske, en ny säkring<sup>4</sup> när felet uppstår så varken löser du problemet eller hindrar att det uppkommer igen eller egentligen vet vad du håller på med alls! Man behandlar inte alla avvikelser på vårdcentralen med plåster, det beror på typen av avvikelse/sjukdom. För vissa sår är plåster utmärkt, men inte för alla, en del måste sys. Och du vill definitivt inte bli behandlad med plåster om du har avvikelsen ”brutet ben” bara för att det brukar fungera eller ”så har vi alltid gjort” eller ”kunderna brukar vara nöjda då” eller ”ingen har klagat hittills”.

### 1.3.1. Slumpfel

En del fel är inte ”pålitliga” i den meningen att de alltid finns där. Somliga kan uppstå en stund efter spänningspåslag, andra kan *försvinna* en stund efter spänningspåslag, eller synbarligen slumpmässigt bara ”då och då”. Detta är den svåraste kategorin fel att hitta. Här gäller det att vara förtrogen med konstruktionen och kunna mäta på rätt ställe, vid rätt tillfälle, för att hitta avvikelserna man är ute efter.

Var ihärdig! Alla fel beror på något, även om detta *något* kan vara svårt att hitta och förstå. Rent slumpmässiga fel är tack och lov sällsynta. Däremot kan man *uppleva* att felet är rent slumpmässiga. Då måste man ställa sig frågan: När uppstår felet? Och dess kompanjon: När uppstår felet **inte**?

Kanske måste man injicera signaler eller störningar kring driftspunkter för att framkalla felet? Kan man dela upp funktionen i mindre delsystem och testa dem var för sig för att tydligare kunna peka ut var felet uppstår?

<sup>4</sup>Eller värre: En kraftigare säkring. Den gamla var ju för svag...

## 1. Konstruera för felsökning

Lämpliga kontrollfrågor när man tror att felet **inte** beror på just X är ”varför?”/”varför inte?”/”är det helt säkert det?”. Först när man är helt övertygad kan man eliminera sitt antagande.

En vanlig källa till slumpfel är glapp på grund av att man inte säkrat ledningar mekaniskt, gjort dåliga lödningar, virningar och andra faststättningar av komponenter.

### 1.4. Konstruera för felsökning

Som vi förstått av ovan kan felsökning vara en tidsödande verksamhet. Om konstruktionen är väl planerad och uppdelad i delsystem underlättar det felsökningen. Det är mycket lättare att leta reda på en felorsak om man kan se vad som händer *mellan* de olika delsystemen. Detta i sin tur betyder att man ska konstruera med tanke på att fel *kommer* att uppstå. Är man planerad och beredd på att fel kommer uppstå blir felsökningen mycket lättare. Har man inte planerat för felsökning kanske felsökning blir omöjlig överhuvudtaget!

Inget utvecklingsarbete går helt som förväntat. Det går helt enkelt aldrig helt enligt planerna. Det gäller alla sorters utvecklingsarbete men är speciellt sant där ett system är så komplicerat att det består av flera delsystem. För att kunna lösa de uppkomna problemen måste man kunna felsöka konstruktionen. Utan felsökning stannar allting upp vid minsta lilla fel. Utan felsökning står man still!

Och även om det just i ditt fall skulle gå utan problem<sup>5</sup> är konstruktion för felsökning outhärligt för att senare också verifiera att konstruktionen fungerar som det var tänkt.

Att felsöka en konstruktion är bland det svåraste man kan göra, men samtidigt kan det vara en mycket stimulerande uppgift. Man måste använda hela sin samlade erfarenhet och till detta en del intuition och logiskt resonemang för att snabbare komma till en lösning. Det kanske kan låta allmänt flummigt men det finns en strategi i flera punkter att använda sig av i en framgångsrik felsökning. Dessa är att bestämma:

1. hur något är fel (symptom),
2. vad det är för detalj som är problemet (identifikation),
3. varför felet uppstod (ursprung),
4. bestämma vad man kan göra åt det (åtgärd) och
5. slutligen hindra att det händer igen (agera proaktivt).

---

<sup>5</sup>Brouhahahahahahaa...

## 1. Konstruera för felsökning

### 1.4.1. Symptom.

För att kunna avgöra om något är fel överhuvudtaget måste man veta hur apparaten är tänkt att fungera i normalfallet. Utvecklar man något för första gången är det naturligtvis svårt att förstå när konstruktionen beter sig avvikande.

En avvikelse från normalbeteendet *kan* indikera ett fel. Det är inte säkert att en avvikelse är ett fel, det kan ju också röra sig om en variation som är inom gränserna för funktionen. Vi lämnar dock dessa mjuka avvikelser åt sidan i det följande.

### 1.4.2. Identifikation.

Att identifiera vad som orsakar symptomet kan vara den mest kreativa processen. Detta steg kräver ofta en ingående kunskap om systemet, dess delsystem och hur de påverkar varann, för att kunna identifiera vad som är felet.

I bästa fall kan man mäta på systemet och man måste då veta vad man mäter på, vilket förväntat mätvärde som finns i mätpunkten och vilka avvikelser som kan tänkas vara acceptabla för att systemet ska fungera. En oacceptabel avvikelse kan vara en första ledtråd i identifikationsfasen.

Förutom de mest enkla systemen kräver identifikationen åtskilliga mätningar och gedigen kunskap om systemet och delsystemens funktion. Identifikationsfasen leder till att man hittar avvikelens ursprung, det kan vara en komponent som är trasig, en programrad som inte gör det man vill, lösa kontakter eller glappande sladdar med mera.

### 1.4.3. Åtgärda.

Åtgärden kan variera. I enkla fall kan det räcka med att byta den felande detaljen. Kanske måste ett kodstycke modifieras eller skrivas om helt. Det kan ju förstås även betyda att man tar ett nytt helhetsgrepp om konstruktionen och kanske tänker om helt angående upplägget.

### 1.4.4. Förhindra.

Den vidtagna åtgärden måste fungera långsiktigt så att just detta fel inte dyker upp igen. Man måste slutligen förhindra att det sker igen. Det kanske inte räcker att byta ut en transistor? Kanske måste en annan transistor med annat  $h_{fe}$ , maximal kollektorström eller effekttålighet användas?

## 1. Konstruera för felsökning

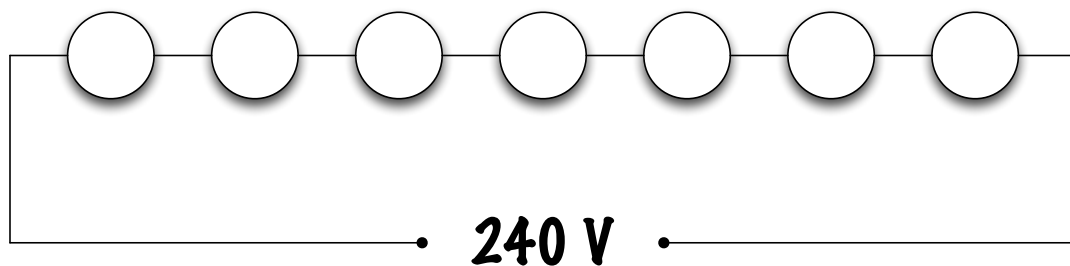
En ändring i ett program får ju inte medföra att fel uppstår på annat ställe eller rutinen bara fungerar på vissa specialfall. Kanske har man gjort samma fel på flera andra ställen i programmet? Med strukturerad programmering och DRY blir det färre ställen i koden som måste ändras. Blir det följdfelet i andra delar av programmet för att man ändrar det som var felet just nu?

Alla dessa steg i felsökningen kan underlättas väsentligt om man konstruerar för felsökning. Att konstruera för felsökning betyder i allmänhet inte att man behöver försvåra själva konstruktionen. I själva verket underlättar det konstruerandet i och med att man försöker isolera olika felkällor från varann. Det kan räcka med att konstruera strukturerat, konstruera modulärt med avgränsade funktioner eller "ansvarsområden" hos olika delsystem. Det innebär också att ha tydliga och entydiga gränssnitt mellan dessa moduler/funktioner/ansvarsområden.

**Exempel** En elektrisk ljusstake med sju ljus lyser inte. Antag att en lampa är trasig och att du har reservlampor, hur hittar du vilken lampa som ska bytas? Symptomet är här klart. Identifikationen är värre. Men vi vet sedan gammalt att om man skruvar ur en lampa slocknar hela ljusstaken. Hur gör du?

När felet lagats är allt dock inte frid och fröjd. Man måste förhindra att felet händer igen, dvs ställa sig frågan "Varför gick den sönder"? Den kan ha passerat sin brinntid, men det kan också vara så att alla lampor inte är lika, vilket medför att någon lampa kan få för stor ström genom sig och därmed brunnit av. För att minska risken att det händer igen kan man alltså extrakolla att alla lampor är av samma sort.

I själva verket är alla lampor seriekopplade i en sådan ljusstake:



**Exempel.** Om det istället är tre lampor trasiga lampor hur gör man då? Uppenbart räcker det inte att byta bara en lampa. Alternativet är att byta tre lampor mot fungerande. Men antalet kombinationer av tre lampor bland sju blir så många att det inte är något intressant alternativ.

Lösningen är att mäta! Använd ett universalinstrument med resistans-mätning och gör mätningar, en på varje lampa tills alla tre är funna.

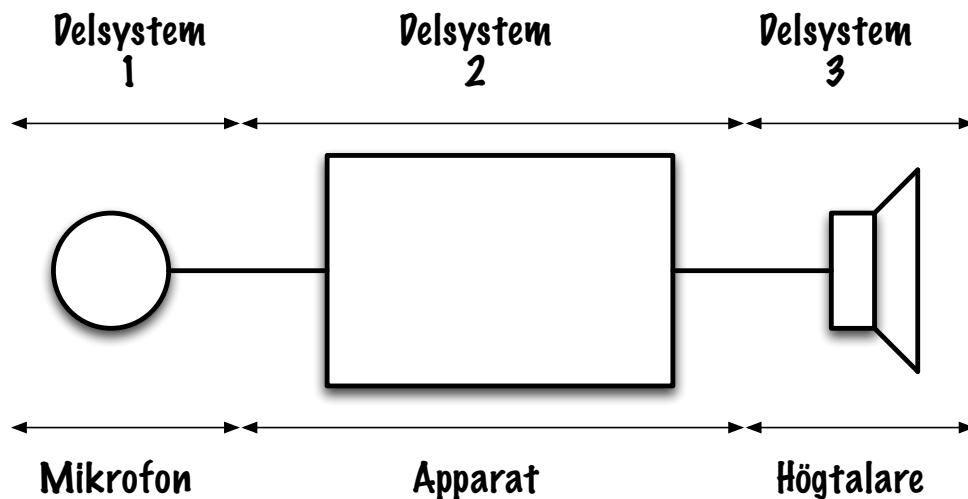


## 1. Konstruera för felsökning

Här kan man också börja fundera på varför lamporna är trasiga? Om en lampa är trasig som i fallet ovan slutar ljusstaken lysa omedelbart, det borde alltså inte kunna vara så att alla tre lamporna gått sönder av överbelastning på en gång. Det verkar osannolikt. Troligen har ljusstaken ramlat i golvet och glödtrådarna skakats av. Minska risken för att detta ska hända igen genom att placera ljusstaken på ett bättre ställe.

För att beskriva hur felsökningen kan gå till kommer här ett större exempel. Det är inte säkert att den använda vägen är bäst, men den visar en strategi. Huvudsyftet är att visa att man måste sönderdela konstruktionen i allt mindre delsystem för att kunna hitta och åtgärda felet.

**Exempel** Man vill utföra digital databehandling på en mikrofonsignal innan dess ljud ska skickas ut till en högtalare. Systemet kan i sin översiktligaste form ritas:



där apparaten innehåller åtminstone en mikroprocessor.

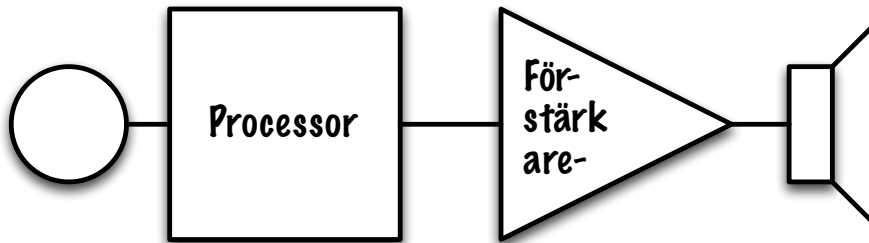
Under utvecklingen konstaterar man att det låter helt förfärligt ur högtalaren när man pratar i mikrofonen. Ljudet är dels svagt, dels mycket sprucket och oangenämt. Man konstaterar att det inte är ett brus ("pschhh") eller ett brum (ett dovt "brrrrrr").<sup>6</sup> Det bara låter väldigt illa helt enkelt. Vad göra?

Utan att veta något om konstruktionen har vi ingen föraning om vad som kan vara problemet. Här börjar felsökningen, vi måste undersöka konstruktionen. Symptomet enligt listan ovan är tydligt, identifieringen kan börja. Eftersom mikrofonsignalen är mycket mindre och därmed svårare att mäta på kan vi felsöka från

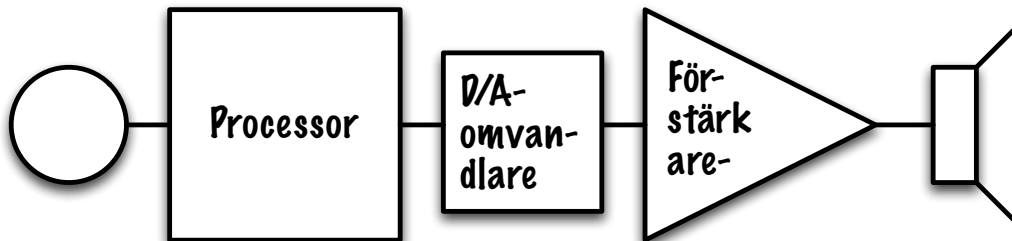
<sup>6</sup>Lär dig särskilja dessa! Ett brum är faktiskt inte detsamma som ett brus, även om många tror det.

## 1. Konstruera för felsökning

högtalaren och bakåt i konstruktionen. För detta måste vi först kontrollera om högtalaren fungerar som den ska och för detta förstår vi att systemet i själva verket måste betraktas som:



eller till och med som:



Här kan vi koppla loss förstärkaringången, kortsluta denna för att ta bort alla insignaler eller lägga på yttre signal från en tongenerator, och lyssna på högtalaren igen. Försvann det störande ljudet?

**Ja:** Nu vet vi att kedjan ”förstärkare-högtalare” är felfria.

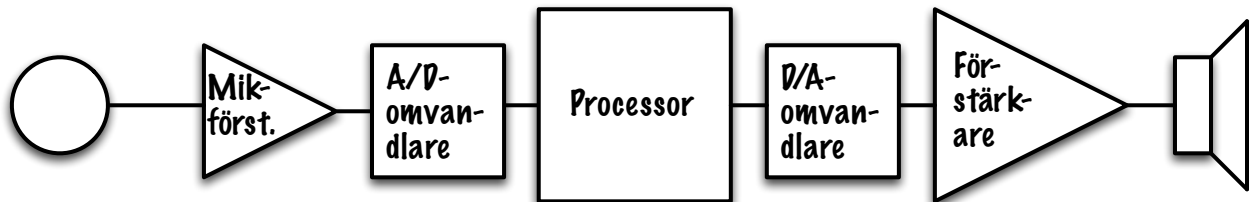
**Nej:** Det störande ljudet skapas någonstans i förstärkaren. Högtalaren i sig är ju en passiv komponent som bara låter som den insignal den har. Fortsätt alltså felsökningen i förstärkaren, dvs betrakta förstärkaren som ett delsystem och felsök detta med kortsluten ingång tills störningen försvinner.

Om svaret var ”Ja” fortsätter felsökningen åt vänster i schemat. Nästa möjliga kandidat är digital-analogomvandlaren. Här kan man prova att ifrån processorn lägga ut olika digitala värden och mäta på omvandlarens utgång för att se om omvandlaren fungerar. Om den inte verkar göra det: Koppla loss den från förstärkaren, det kan ju vara förstärkar-*ingången* som påverkar.

Ett annat prov är att generera en ton i processorn och höra om den återges välljudande. I den här fallet valde man att gissa att processorn gjorde det man trodde och fortsatte till kopplingen mellan mikrofon och processor. För att processorn ska

## 1. Konstruera för felsökning

läsa in ljudet måste den analog-digitalomvandla (A/D-omvandla) mikrofonsignalen, så vårt blockschema utökas med det nya delsystem, dels A/D-omvandlaren men också en mikrofonförstärkare som också redan finns på plats:



Felsökningen kan fortsätta med att bryta upp kopplingen mellan mikrofonförstärkaren och A/D-omvandlaren, lägga på en sinussignal som insignal och lyssna igen. I det här fallet låter det betydligt bättre med egen pålagd signal. Alltså borde hela kedjan mellan A/D-omvandlare och högtalare vara oskyldig här.

Återstår mikrofonen (som sällan går sönder) och mikrofonförstärkaren. Mätning med oscilloskop på signalen före respektive efter förstärkaren visar att signalen ser distorderad och klippt ut efter förstärkaren.

Problemet visade sig till slut vara att mikrofonen som utsignal gav en växelspanning runt 0 V, dvs även negativa halvperioder under 0 V fanns med i insignalen. Bara de positiva halvperioderna kom dock vidare till A/D-omvandlaren och således behandlades bara de positiva halvperioderna i fortsättningen med distorsion som följd. Lösningen var att lägga till en kopplingskondensator i serie med mikrofonsignalen innan denna påfördes förstärkaren. Detta gjorde att mikrofonen inte DC-mässigt kortslöt förstärkaringången till jord. Då lät det genast bättre. Felsökningen klar. Problemet löst.

Sökmetoden ovan var ”sökning medelst uppdelning”. En självklar metod om systemet redan är uppdelat i funktionella enheter. Om vi konstruerat för felsökning är detta redan gjort och felsökningen underlättas omedelbart. Det är dessutom enkelt att sätta in testsignaler och utföra mätningar i gränssnitten mellan enheterna. Om en signal finns på ena sidan om ett delsystem men inte på andra så är det lätt att sluta sig till var signalen försvann.

### Fundera på:

- Tänk på ett tal mellan 1 och 256. Låt en kompis gissa vilket tal det är. Du får bara svara att hans gissning är ”för hög” eller ”för låg”, eller ”rätt”. Hur många gissningar behövs?
- Gör som ovan fast låt talet vara mellan 1 och 1024. Hur många gissningar behövs nu?

### 1. *Konstruera för felsökning*

- Tänk att du har ett system som består av 256 delsystem så att det ena delsystemets utsignal är det andra delsystemets insignal och så vidare. Det är fel i ett av delsystemen! Hur kan du snabbast hitta vilket delsystem som är trasigt?

# A. Citatsamling

Här kommer en liten citatsamling som på olika sätt beskriver det som detta häfte handlar om:

- Något är fel när vi inte får ett väntat resultat
- Alla fel beror på något
- Det är sällan fel på komponenterna
- Utan felsökning står man still
- Är det värt att göra är det värt att göra i en subrutin
- "Make everything as simple as possible, but not simpler." — Albert Einstein
- Man löser problem **bättre** och man felsöker **bättre** om man vet **mer** om systemet än om man vet precis så mycket som behövs för att lösa uppgiften. — Gammalt Forth-ordspråk
- "The enemy of a good plan is the dream of a perfect plan." — Carl von Clausewitz
- "Never, never, never give up!" — Winston Churchill