How GUIDs Will Ruin Your Life

Mark S. Rasmussen

improve.dk

C:\>whoami

- Technical Lead @ iPaper

- Developer

- DBA

- Sysadmin

- Project lead

- Comp. Sci. @ AU

GUID – huh?

- UUID
  - *A **Universally Unique Identifier** (UUID) is an identifier standard used in software construction, standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE).*
    - http://en.wikipedia.org/wiki/Universally_Unique_Identifier

- GUID
  - *The term GUID also is used for Microsoft's implementation of the Universally Unique Identifier (UUID) standard.*
    - http://en.wikipedia.org/wiki/Guid

1337

{21EC2020-3AEA-1069-A2DD-08002B30309D}

int
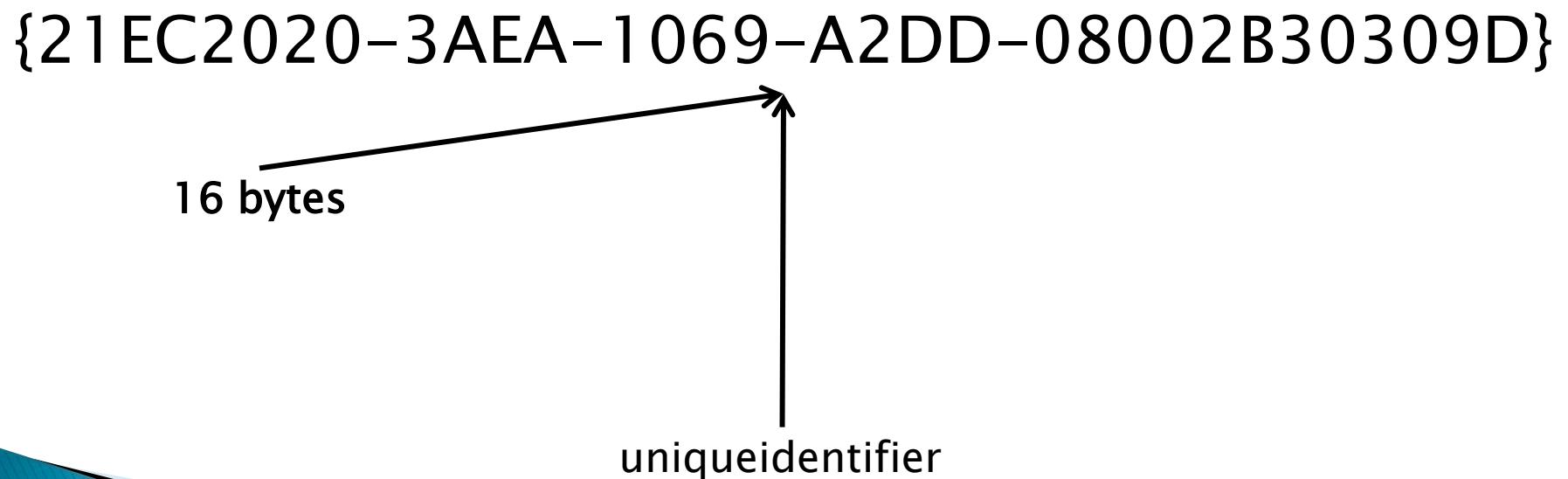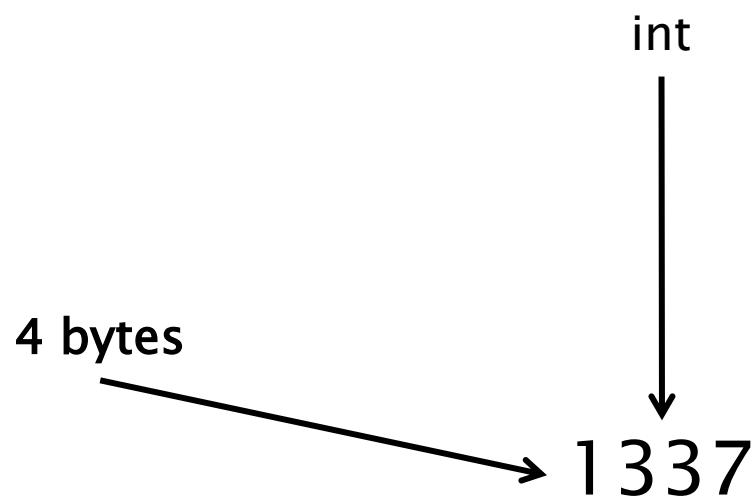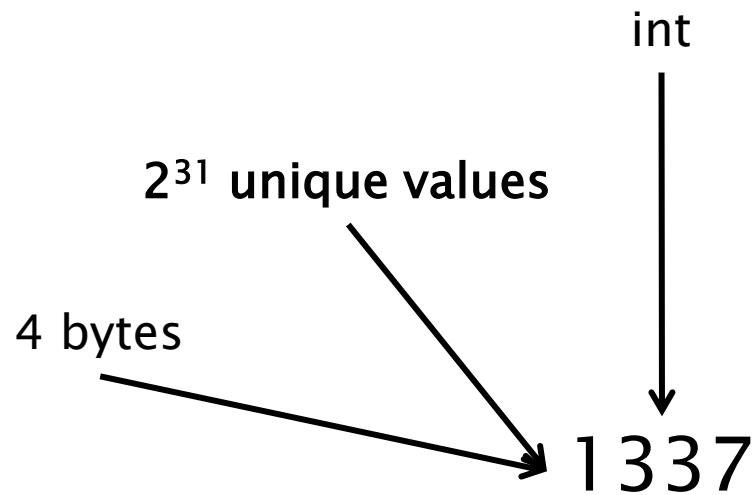
1337

{21EC2020-3AEA-1069-A2DD-08002B30309D}
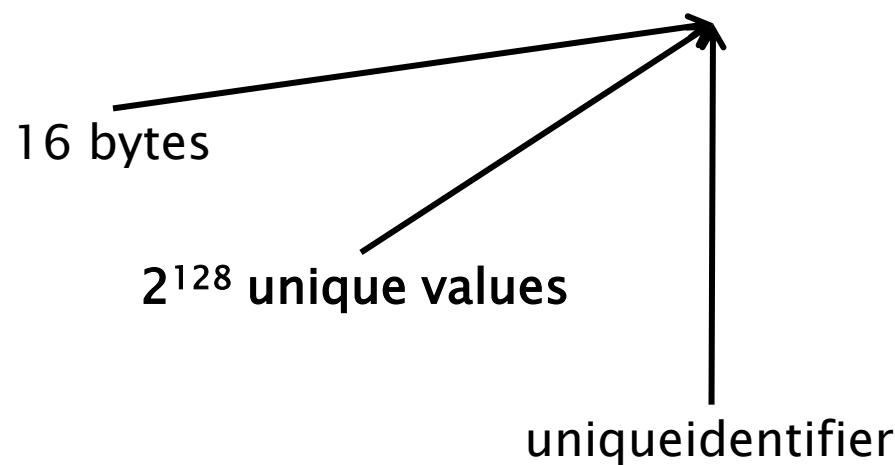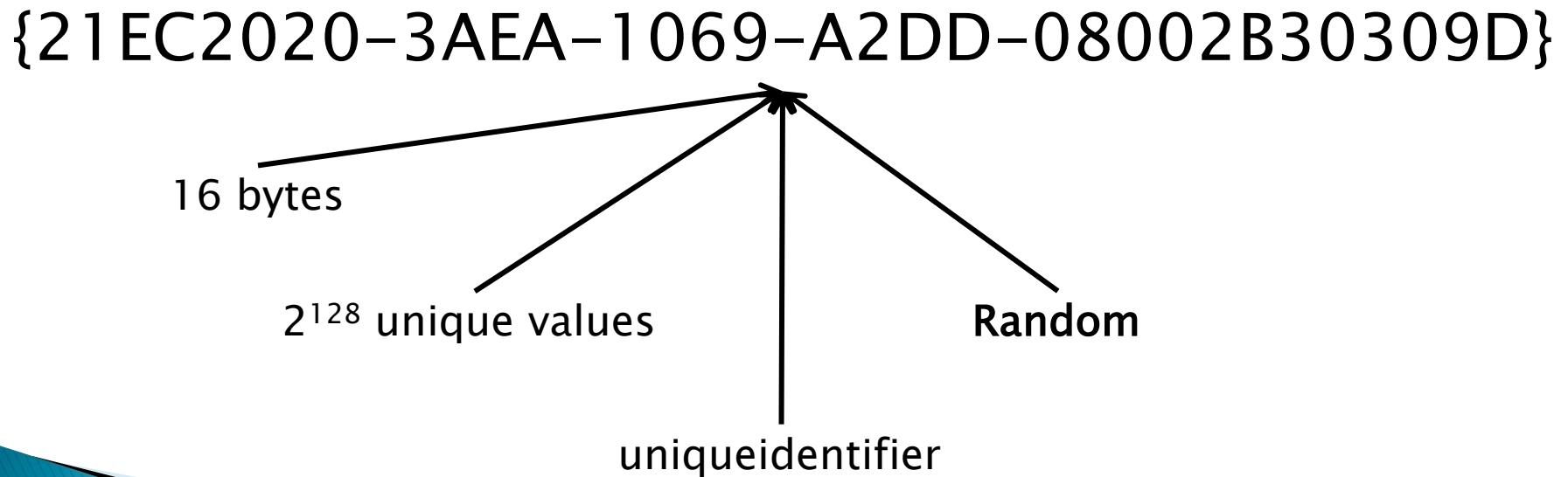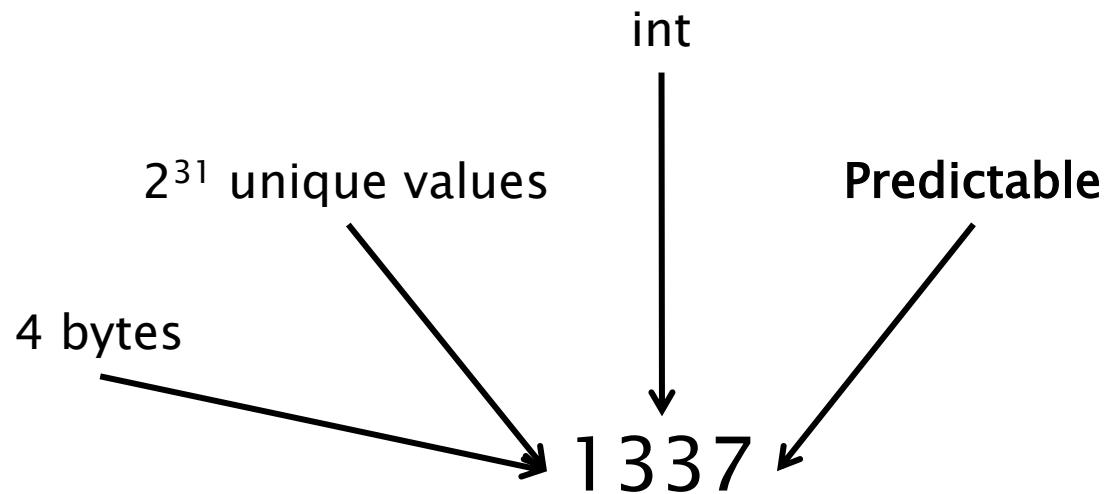
uniqueidentifier
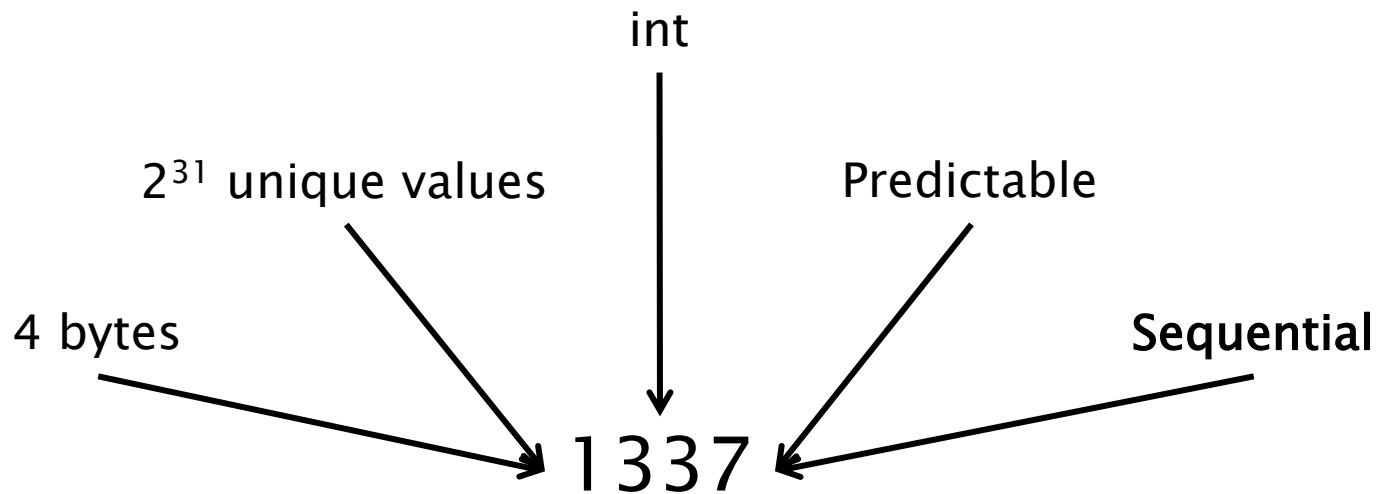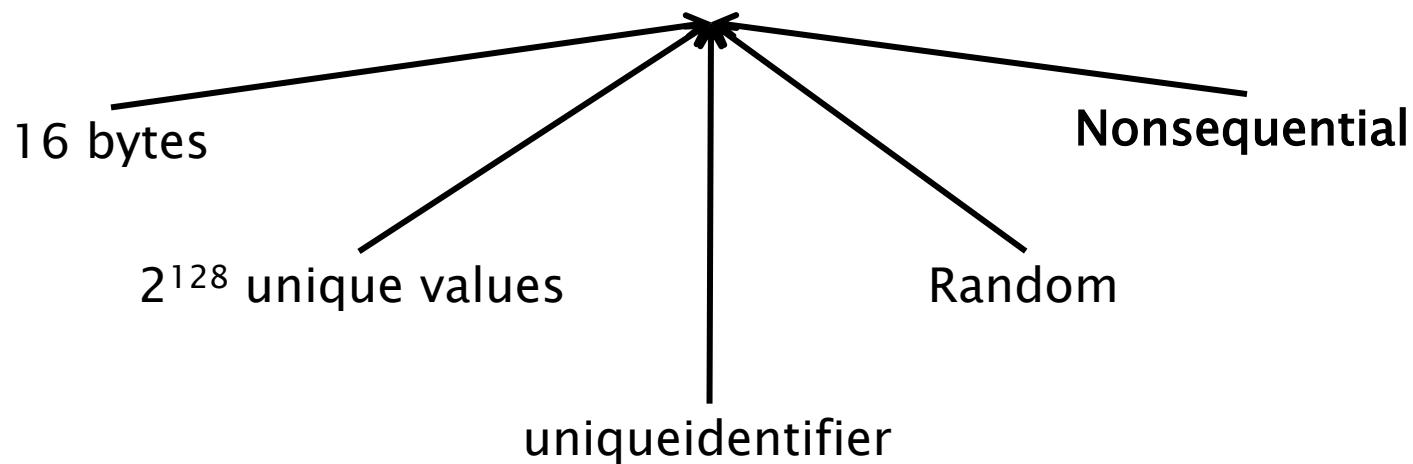
int

4 bytes

1337

{21EC2020-3AEA-1069-A2DD-08002B30309D}

16 bytes

uniqueidentifier

int

2^{31} unique values

4 bytes

1337

{21EC2020-3AEA-1069-A2DD-08002B30309D}

16 bytes

2^{128} unique values

uniqueidentifier

int

2^{31} unique values

Predictable

4 bytes

1337

{21EC2020-3AEA-1069-A2DD-08002B30309D}

16 bytes

2^{128} unique values

Random

uniqueidentifier

int

2^{31} unique values

Predictable

4 bytes

Sequential

1337

{21EC2020-3AEA-1069-A2DD-08002B30309D}

16 bytes

Nonsequential

2^{128} unique values

Random

uniqueidentifier
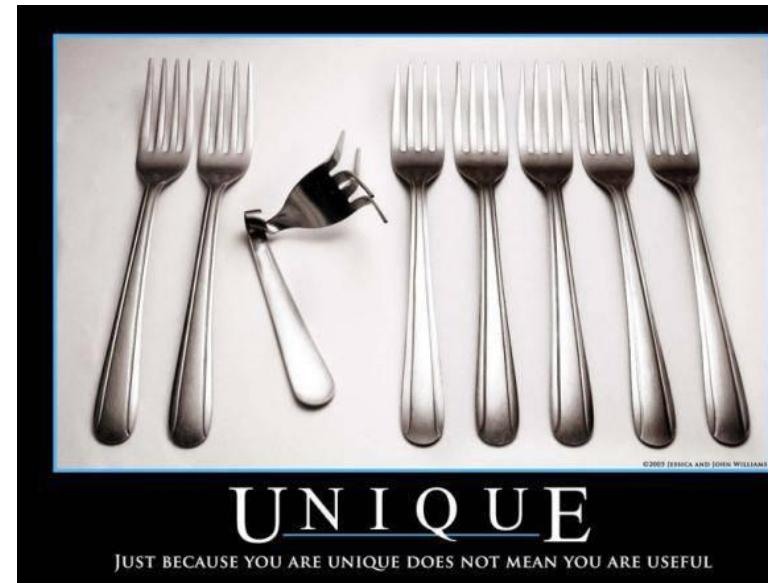
Cardinality

- 2^{31}
  - ~2 billion

- 2^{128}
  - Amounts to "Roughly 2 trillion per cubic millimeter of the entire volume of the Earth."

Unique & unpredictable

▸ Client side generated PKs

▸ No need for centralized generation

▸ Simplifies / optimizes app code

▸ Security by obfuscation

UNIQUE

JUST BECAUSE YOU ARE UNIQUE DOES NOT MEAN YOU ARE USEFUL

4 vs 16 bytes

▸ +12 bytes for all nonclustered indexes

▸ Imagine this…

Cluster type	Width	Rows	NC indexes	MB
INT	4	100,000,000	8	3.051 MB
BIGINT	8	100,000,000	8	6.103 MB
GUID	16	100,000,000	8	12.207 MB

4 vs 16 bytes

- Affects DML
- Memory waste
- Expensive joins
- Backups
- Maintenance

Nonsequential

▸ Often used as PKs

▸ SSMS will, by default, create a clustered index on the PK

▸ GUIDs as clustered keys – Demo

Page splits

Page 1	Page 2		Page 10293
1	13		239403
4	16		247840
7	18	···	252034
8	20		384039
10	21		424003

INSERT INTO … VALUES (5)

Page splits

Page 1	Page 3		Page 10294		Page 2
1	13		239403		7
4	16	...	247840	...	8
5	18		252034		10
	20		384039		
	21		424003		

Do we have a problem?

Performance counters

```
SQLServer:Access Methods
    Full Scans/sec                                          88,907
    Index Searches/sec                                 121.638,232
    Page Splits/sec                                        435,546
```

```
\\VENETIAN
    PhysicalDisk                                            _Total
        Avg. Disk Queue Length                             102,827
        Avg. Disk sec/Read                                   0,100
        Avg. Disk sec/Write                                  0,009
        Disk Reads/sec                                     883,066
        Disk Transfers/sec                               2.627,220
        Disk Writes/sec                                  1.744,154
        Split IO/Sec                                         0,000
```

What can we use instead of GUIDs?

- Range limitation is likely not an argument

Type	Size	Max value
TINYINT	1	255
SMALLINT	2	32,768
INT	4	2,147,483,647
BIGINT	8	9,223,372,036,854,775,807
DECIMAL(38,0)	17	99,999,999,999,999,999,999,999,999,999,999,999,999

- Sequential GUIDs
  - Can be predicted
 - Application responsibility
  - Increases risk of duplicates
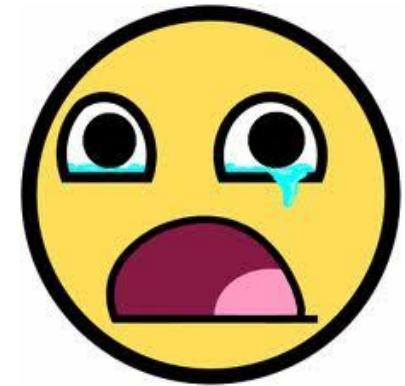
Demo

What if we have to use GUIDs?

▸ Reduce FILLFACTOR Demo

▸ Use NEWSEQUENTIALID() instead of NEWID()

▸ Primary key doesn't have to be on the clustered key

▸ Can we change the database schema?
  ◦ Create new clustered indexes
    · Even if they're not used
    · Sort order shouldn't matter if using GUIDs before
    · Optimize joins
    · Optimize foreign keys
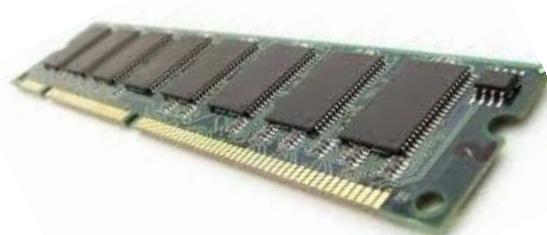
But my clustering key is a primary key!

- Database will have to go offline

- Disable FKs & nonclustered indexes

- Drop the PK & create new clustered index

- Create nonclustered PK on GUID column

- Reenable FKs (with CHECK!) & nonclustered indexes

- Go online!

Can't change schema?

▸ Monitor and maintain indexes

▸ Buy more hardware

# And the worst part?
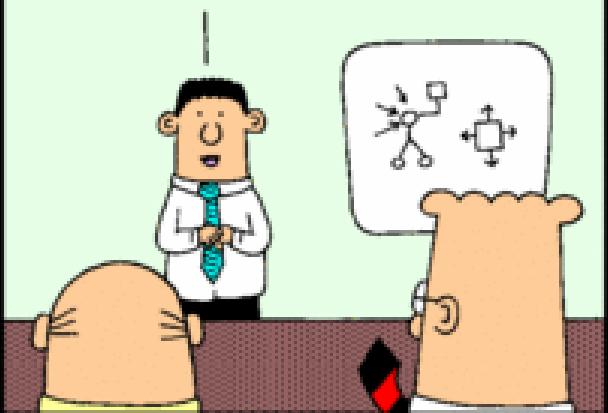
3400 AD
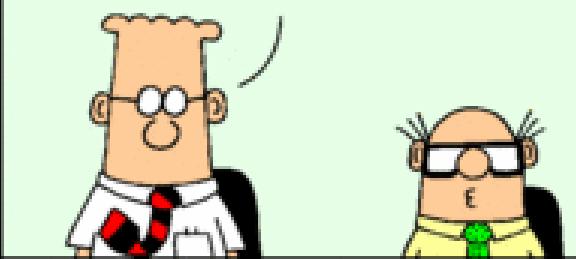
Key takeaways

- Choose your clustering key wisely

- PK != CK

- Storage is rarely the problem

- There are many alternatives

- Monitor & maintain
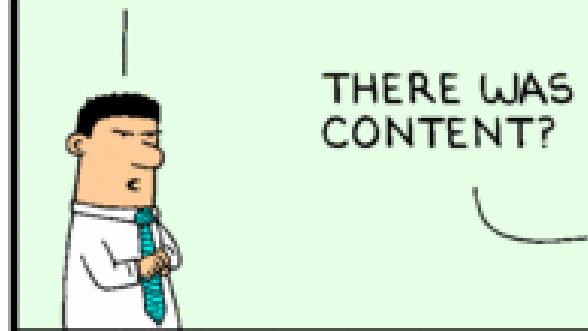
Mark S. Rasmussen
improve.dk
@improvedk