

# Självständigt arbete

-

Datateknik

**Utvärdering av arbetsprocess och CMS**

**Douglas Pettersson**



**Mittuniversitetet**

MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. Campus Sundsvall Holmgatan 10, SE-851 70 Sundsvall.

Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.

Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

## Sammanfattning

I dagens webb används ofta många olika innehållshanteringssystem för att göra det enkelt för användare att hantera deras egna hemsidor. I utvecklingsarbetet hos webbyråer bedrivs även någon typ av projektledning. Syftet med denna undersökning är att utvärdera och analysera ett företags befintliga CMS och deras arbetsprocess. Denna utvärderingen kommer göras utifrån ett visst antal kriterier som är kännetecken för olika projektledningsmetoder samt en separat utvärdering för ett passande CMS utifrån specifikationer och intervjuer. Här kommer faktorer som licens, funktionalitet, plattformstöd, vidareutveckling, popularitet samt exempelvis programmeringsspråk. Denna undersökning visade på att företaget arbetar med ett agilt arbetssätt, samt att deras befintliga CMS-system är det optimala för deras verksamhet för att de ska upprätthålla sina möjligheter att helt skräddarsy lösningar och ha full kontroll på produkten.

**Nyckelord:** Projektledning, CMS, .NET , C#.

## Abstract

On today's web it is very common to be using different sorts of content management systems to manage the information on their own websites. In the development process most web agencies use some type of project management method. The purpose of this analysis is to evaluate and analyze the business current CMS and their software development process. This evaluation will be made from a certain amount of criteria that are known and specific features of different project management methods and also a separate assessment to see if there is another fitting CM-system from interviews and the different systems specifications. Factors such as licensing, functionality, platform support, expansions, popularity and also the programming language of which it builds on. This review showed that the company is using an agile software development process to develop their products as well as seeing that their existing content management system made by their selves is the most optimal seeing their need to create custom solutions and having full control of the product.

**Keywords:** Project management, CMS, .NET, C#.

# Innehållsförteckning

<b>Sammanfattning</b> .....	<b>ii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Terminologi</b> .....	<b>vi</b>
<b>1 Inledning</b> .....	<b>1</b>
1.1 Bakgrund och problemmotivering.....	1
1.2 Övergripande syfte.....	2
1.3 Konkreta mål .....	2
1.4 Avgränsningar.....	2
<b>2 Teori / Bakgrundsmaterial</b> .....	<b>3</b>
2.1 ASP.NET.....	3
2.2 C#.....	4
2.3 Umbraco .....	5
2.4 DotNetNuke (DNN).....	7
2.5 Kentico.....	8
2.6 Mindbite SiteFactory.....	10
2.7 Agil Projektledning / Systemutveckling.....	11
2.8 Scrum.....	13
2.9 Vattenfallsmodellen.....	16
2.10 Fördelar och Nackdelar.....	18
<b>3 Metod</b> .....	<b>23</b>
3.1 Övergripande metod.....	23
3.2 Kriterier hos CMS.....	23
3.3 Undersöka passande CMS:er.....	23
3.4 Utvecklingsmöjligheter.....	24
3.5 Utvärdera & föreslå CMS och övriga system.....	24
3.6 Samla fakta kring arbetsprocessen.....	24
3.7 Utvärdera arbetsprocess.....	24
3.8 Analys & slutsatser .....	24
3.9 Utvärderingskriterier.....	25
<b>4 Resultat</b> .....	<b>27</b>
4.1 SiteFactory funktionalitet.....	27
4.2 CMS.....	29
4.3 Arbetsmetodik.....	35
4.4 Slutsatser från utvärdering av CMS.....	38
4.5 Slutsatser från utvärdering och analys av projektmetodik.....	38
4.6 Rekommenderade åtgärder.....	39
<b>5 Slutsatser</b> .....	<b>40</b>
5.1 Tillförlitlighet hos utvärderingen utav CMS.....	41
5.2 Tillförlitlighet hos utvärdering/analys av projektmetodik.....	41
5.3 Förslag till vidareutveckling och forskning.....	42

<b>Källförteckning.....</b>	<b>43</b>
<b>Bilagor.....</b>	<b>47</b>

# Terminologi

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
CMS, CM-System	Content Management System
.NET	Microsoftutvecklat mjukvaruramverk
C#	Objektorienterat programmeringsspråk inom .NET
Scrum	Agil projektledningsmetodik

# 1 Inledning

## 1.1 Bakgrund och problemmotivering

I dagens bransch inom webb och systemutveckling där det är hög konkurrens om marknadsandelar så krävs det för företagen att de kan arbeta på ett så effektivt sätt som möjligt för att samla kunder på ett bra sätt samtidigt som de ska vara konkurrenskraftiga på marknaden gällande både prissättning och hur fort man kan gå från första möte till färdig produkt.

Viktiga delar i detta är hur företaget arbetar med projektledning och deras arbetsprocesser där det finns många varianter av hur man vill arbeta och sätta upp sina utvecklingsprocesser för att vara så effektiva som möjligt. Vad företagen använder för system i form av hjälpmedel eller *CMS-system* (Ett form av system för att hantera innehåll på t.ex. webbplatser och intranät) är tillsammans med projektledningen två av de allra viktigaste kriterierna att kunna förbättra eller att ha förbättrat för att uppnå optimala resultat.

För att göra detta arbetar nu för tiden många system- och webbutvecklingsföretag med agila utvecklingsmetoder för att arbetet som genomförs ska få bästa möjliga resultat och göra det mesta av de medarbetare som finns. Det finns idag många möjligheter gällande hjälpmedel och CMS-system som företagen kan arbeta med, och likaså med arbetsmetoder som ofta skiljer sig åt på flera olika sätt även att de ofta är lika i grunden. Inom denna studie ute på företaget kommer det undersökas hur företagets arbetsprocess ser ut i dagsläget, hur den kan förändras samt mot vilken typ av projektledning den främst är riktad mot.

Då en del av hela utvecklingsprocessen även till stor del består av CMS och andra hjälpsystem i dagsläget så kommer även utvärderingar av andra CMS samt företagets behov av övriga hjälpmedel eller system genomföras.

## 1.2 Övergripande syfte

Målet med denna rapport är att genom utvärderingar och analyser av arbetsprocesser och system redogöra för företagets eventuella utvecklingsmöjligheter gällande deras arbetsprocess och system och se över om deras befintliga system är deras bästa val eller om det finns fler möjligheter för företaget. Gällande arbetsprocessen så är syftet att se över hur deras arbetsmetodik ser ut i dagsläget och hur den eventuellt kan förbättras & mot vilken typ av systemutvecklingsmetodik den ställer sig främst.

## 1.3 Konkreta mål

Frågeställningar som kommer att besvaras i detta arbete är:

- Varför använder Mindbite det nuvarande CMS:et? Samt vilka funktioner etc. det innehåller
- Vilka övriga CMS finns det på marknaden inom C# med .NET?
- Finns det vidareutvecklingsmöjligheter mot nyare tekniker eller är det lämpligare att byta ut nuvarande CMS mot ett annat?
- För & Nackdelar med att behålla det nuvarande samt ersätta mot ett nyare?
- Vilken typ av systemutvecklingsmetoder använder sig företaget främst sig av i dagsläget och hur kan deras processer eventuellt förbättras?

## 1.4 Avgränsningar

De CMS-system som kommer att tas upp i denna rapport är endast system som bygger på och är kompatibla med *Microsoft .NET* (Ramverk för programmering från Microsoft) då det är detta företags utvecklare främst är utbildade och kunniga inom. Jag kommer även endast att jämföra och dra paralleller med de mest använda arbetssätten bland systemutveckling, varav dessa är vattenfallsmetoden samt den agila metoden (Scrum).



## 2 Teori / Bakgrundsmaterial

I webbindustrin finns det hundratals med CMS-system däribland ett par stycken är de allra populäraste. Webbyrån Mindbite AB där detta examensarbete genomförs valde att utveckla ett eget CMS – SiteFactory på grund av att ha tillgång till sitt eget system med alla utvecklingsmöjligheter som då fanns, samt även flexibiliteten det erbjuder men även den extra supporten de kan erbjuda sina kunder genom att kunderna från direktkontakt gratis med de som har byggt upp systemet och dess funktioner. På Mindbite arbetar de främst med ASP.NET Webforms med C# som back-end programmeringsspråk.

För att ge en inblick i vad det finns för olika CMS:er som är baserade på .NET så ska jag i denna del gå igenom 3 utvalda CMS-system utifrån de allra populäraste .NET CMS:er som finns på marknaden idag. De tre jag har valt ut till denna undersökning är DotNetNuke (DNN) samt Umbraco och Kentico där alla dessa tre är med bland de topp 5 populäraste .NET-baserade CMS-systemen idag.<sup>1</sup>

För att få en bra grund till att förstå begrepp och arbetssätt inom de olika projektmetodikerna kommer jag även att ta upp bakgrundsfakta allmänt kring det agila arbetssättet samt även den traditionella vattenfallsmetoden.

### 2.1 ASP.NET

.NET kan man beskriva som en plattform för flera olika tekniker inom applikations- och webbutveckling. Plattformen är byggd och underhålls utav Microsoft. .NET släpptes först tidigt år 2002 som version 1.0 och är idag uppe i version 4.7 tillsammans med Visual Studio 2017 som är det verktyg man bedriver utveckling inom .NET i, denna version släpptes den 4 april 2017.

ASP.NET och Visual Studio har mycket inbyggd funktionalitet för hög prestanda, minneshantering, säkerhet, portabilitet samt deras egna CTS (*Common Type System*) som hela tiden visar möjliga datatyper och kodbitar som är kompatibla med varandra, och beskriver även varför något inte funkar om något fel finns, detta är mycket hjälpsamt vid utveckling i .NET i VS.

Man har tidigare i .NET använt sig mycket utav .NET Web Forms som är en del utav hela ramverket. Under senare år har detta börjat fasas ut mot .NET MVC (*Model View Controller*). MVC gör det enklare att hantera komplexa lösningar genom att de olika delarna utav applikationen delas upp i både models och controllers för att hantera och visa data i sin applikation eller webbplats. MVC passar även mer för utvecklare som vill ha en större kontroll över

---

1

applikationens beteende där t.ex. Web Forms använder sig utav serverbaserade formulär för att hantera data.<sup>2 3</sup>

## 2.2 C#

C-sharp (C#) är ett objektorienterat programmeringsspråk.<sup>4</sup> Språket utvecklades av Microsoft till .NET och blev senare ett standardiserat språk utav ECMA (*ECMA är en organisation som grundades för att standardisera datorsystem i hela Europa | European Computer Manufacturers Association*).<sup>5</sup> Språket är i grunden baserat på C++ och har därmed många likheter med detta. C# är även ett programmeringsspråk som inte är beroende utav en viss plattform, utan fungerar på även Mac och Linux t.ex.<sup>6</sup> Dock har Microsofts utvecklingsverktyg tidigare endast funnits till Windows, men det har nu ändrats med introduktionen av Visual Studio 2017 samt .NET Core som är till stor del ett plattformsoberoende framework för utveckling av applikationer och webbsidor för både Mac, Linux & Windows.<sup>7</sup>

C# är det språk som i regel används back-end för att utveckla funktionaliteten bakom .NET applikationer och webbsidor, därav är detta även det programmeringsspråk som de CMS som valt att analysera är skrivet i, dvs. all funktionalitet bakom dessa CMS-system är uppbyggt i detta språk.<sup>8</sup>

## 2.3 Umbraco

Umbraco är det näst populäraste av de tre och ligger i dagsläget som grund för en bit över 400.000 installationer. CMS:et är ett open-source CMS som används till grund för både webbsidor och applikationer men även exempelvis intranät för att sköta informations- och innehållshantering i dessa olika typer av system eller webbsidor. Umbraco är skrivet i C# vilket är det språk Mindbite arbetar med, vilket till en grund gör detta till ett bra jämförelseexempel. Sedan version 4.5 går Umbraco under en MIT-licens<sup>9</sup>. Detta innebär att produkten helt gratis kan ladda ner programvaran och använda den utan restriktioner gällande varken användning, kopiering, vidareförsäljning, sammansättningar eller publiceringar.<sup>10</sup> I och med att Umbraco inte har någon tvångskostnad för att använda programvaran och dess dokumentation så är det ett väldigt effektivt verktyg både gällande kostnad och anpassningsbarhet då det även bygger på .NET vilket ger stora friheter gällande både funktionalitet och design.

Även om Umbraco är fritt att använda erbjuder de även betalversioner med automatiska uppdateringar och oändlig hostingservice för webbsidor, detta

---

2  
3  
4  
5  
6  
7  
8  
9  
10

kostar från €25 för ”Starter” och uppåt tills det att man får en skräddarsydd lösning och prissättning för just dina behov i form av ”Enterprise”-versionen.

Starterversionen ger dig automatiska uppdateringar, obegränsad lagring och bandbredd, tillgång till en Umbracosida samt Umbraco TV och Umbraco Forms för att nämna endast ett par delar. Det ingår alltså ingen support exempelvis för basversionen utan för detta får man uppgradera till ”Professional” vilket erbjuder all funktion som Starter ger dig fast med garanterad 99,9% ”uptime”, support via chat och responstid inom 24 timmar. Enterpriseversionen ger dig lite mer utöver detta och ger dig ett antal sidor som skräddarsys och prissättningen sätts efter detta, men även support via telefon, 2 timmars responstid via chat eller mail samt rådgivning av kod. Själva CMS:et har flera inbyggda funktioner, däribland en *WYSIWYG-editor* (En editor som man enkelt kan använda för att autogenerera CSS och HTML-kod, i vissa fall även Javascript), mediahantering, medlemshantering, flerspråksstöd, olika användartyper, hantering av skyddade sidor, tidsstyrd publicering etcetera.<sup>11</sup>

Här ingår bland annat friheten att kunna använda både .NET Webforms och .NET MVC tillsammans. MVC som av dessa två är den senare tekniken inom språket som släpptes år 2008 då Web Forms till viss del börjar fasas ut utav Microsoft och inte är kompatibelt med .NET 5.0, dock fortfarande kompatibelt med .NET plattformen i övrigt.<sup>12</sup>

Umbraco sparar data i relationsdatabaser, i de allra flesta fall så används Microsoft SQL Server som databas men sedan Umbraco 4 så finns det inbyggt stöd för både MySQL och MSSQL då stödet för VistaDB togs bort vid Umbraco version 4.6.<sup>13 14</sup> Systemet fungerar även på Microsoft IIS som server. Umbraco's back-end språk är som tidigare nämnt C# och dess front-end bygger på Microsoft .NET. Umbraco använder standardiserade funktioner från .NET som t.ex. 'mastersidor' som används i syfte för att kunna återanvända sidlayouter och design på ett enkelt sätt.<sup>15</sup>

Det vanliga upplägget för Umbraco är att systemet installeras på IIS Express där webbplatsen eller intranätet sedan körs vilket är det allra smidigaste sättet för att installera och köra Umbraco.<sup>16</sup>

Mindbite använder sedan tidigare Umbraco i mindre utsträckning när det finns behov utav ramverket MVC till ASP.NET då det finns vissa kunder som efterfrågar just detta eller funktionalitet som kräver MVC. Mindbite SiteFactory har alltså inget stöd för ASP.NET MVC utan det bygger i grunden på ASP.NET Web Forms.

---

11  
12  
13  
14  
15  
16

Umbracos krav för att driva en Umbracoinstallation av den senaste relevanta versionen Umbraco 7.X.X är IIS 7.0 och uppåt, SQL Server 2008 eller bättre, ASP.NET 4 och uppåt alternativt MySQL.<sup>17</sup>

## 2.4 DotNetNuke (DNN)

DNN som det förkortas är likt Umbraco ett CMS-system baserat på Microsoft .NET och är skrivet i programmeringsspråket C#. DNN plattformen är open source och är licenserat under MIT licensen vilket är samma licens som Umbraco använder sig av.<sup>18</sup> DotNetNuke används på över 750.000 webbsidor och applikationer över hela världen och har mer än 1 miljon registrerade användare i deras community vilket gör det till ett väldigt stort nätverk med tillgång till support från andra medlemmar.<sup>19</sup>

CMS:et distribueras både som en gratisversion under MIT licensen men finns även tillgänglig som DNN Evoq Content / Basic / Engage som inte är open source utan istället är betalversioner med mer funktionalitet och support.<sup>20</sup> Dessa betalversioner har olika kostnadsnivåer och varierar från \$2,999/år för Evoq Basic till \$7,999/år för Evoq Content och slutligen från \$19,999/år för Evoq Engage. Samtliga av dessa olika produktversioner har funktioner för exempelvis Visual Studio Templates, WebAPI, inbygga Javascriptbibliotek, återanvändbara controllers och designer, flersidshantering, avancerad URL-hantering, rollbaserad säkerhet samt inbyggt Gogole Analytics för att nämna de viktigaste. Den grundläggande Evoq Basic innehåller något mer funktionalitet än ovan nämnda som exempelvis inbyggda responsiva teman för mobila enheter, moduldelning mellan flera webbsidor, sökfunktionalitet för webbsidan, versionshantering, stöd för Microsoft Azure och även obegränsad mail-support samt tillgång till DNN's onlinetillgängliga dokumentationscenter.

Evoq Content och Evoq Engage har funktionalitet långt över detta som exempelvis formulärbyggare, inbyggd integration med t.ex. Dropbox, innehållsanalyserare, samt mycket inbyggt stöd för tredjepartssystem som Facebook eller MailChimp och obegränsad teknisk support.<sup>21</sup>

Då Mindbite främst fokuserar på att ge deras kunder ett kostnadseffektivt sätt att publicera och hantera deras webbsidor så kommer vi av dessa fokusera på DNN Platform som är helt fritt att använda.

DNN Platform är möjligt att vidareutveckla och addera funktionalitet till som redan finns i de kostnadsbaserade versionerna. Man kan antingen köpa eller ladda ner fria färdiga moduler till systemet eller även utveckla sina egna. I princip all funktionalitet som redan finns i CMS:et kan bytas ut. CMS-systemet har även inbyggt säkerhet på en hög nivå som hela tiden testas för att uppfylla

---

17  
18  
19  
20  
21

de allra hårdaste säkerhetskraven.<sup>22</sup> DNN erbjuder även ett stort forum med svar på allt man kan tänka sig stöta på för problem, allt från hur man grundläggande installerar systemet till hur man bygger egna tillägg.<sup>23</sup> Under tidigare år har DNN endast varit kompatibelt med och bygger på .NET Web Forms men har likt Umbraco även stöd för att använda Model View Controller (ASP.NET MVC) som haft en stark tillväxt sedan introduktionen av version 1.0 år 2009.<sup>24</sup>

DotNetNuke är endast kompatibelt med och har som krav att minst SQL Server 2008 skall vara installerat, övriga krav är minst .NET 4.0 och IIS 7 och uppåt samt Windows Vista som operativsystem. DNNSoftware rekommenderade miljö är Windows Server 2012 R2, IIS 8.5, SQL Server 2012 samt .NET Framework 4.5.<sup>25</sup>

## 2.5 Kentico

Kentico tog jag med som ett alternativ som är bland de största .NET-baserade CMS-systemen och driver idag cirka 25.000 webbplatser i 100 olika länder.<sup>26</sup> Kentico använder sig utav Microsoft ASP.NET och Microsoft SQL Server och är precis som de andra två CMS:erna skrivet i C#. Kentico är kompatibelt med Microsoft Azure plattformen<sup>27</sup> och tillåter utveckling via Kenticos egna motor eller via Visual Studio.<sup>28</sup>

Kentico finns både som gratis programvara och som betalversion. Kentico CMS Free Edition har dock vissa begränsningar, antal sidor och dokument får uppgå till max 1.000st, antal sidmedlemmar max 100st, administratörer och editörer får man ha en utav varje samt att man har som krav på sig att ha länk till Kentico i sin sidfot på alla sidorna på webbplatsen - ”Powered by Kentico CMS”. Man får heller inte i denna version någon support utan man är helt beroende på Kentico's forum.<sup>29</sup> Det finns flera olika versioner att välja på, Kentico med 1 webbsidelicens, 1 serverlicens för 10 sidor samt 1 serverlicens för obegränsat antal sidor. De två sistnämnda serverlicenserna kräver dock att minst 50% av alla sidorna är kontrollerade av företaget/organisationen som äger serverlicensen. Kentico Base börjar på en kostnad av \$4,499 och till detta ingår ett års underhåll i form av fria uppgraderingar och teknisk support. Priserna går upp emot \$399,000 för ”Unlimited Server License” för Kentico EMS Corporate.<sup>30</sup>

Kentico CMS Base ger grundläggande CMS-funktionalitet med möjlighet att ha flera sidor på webbplatsen or erbjuder även 24/7 support. Basversionen har även E-handel inbyggt samt delar av marknadsföring, intranät samt skalbar

---

22  
23  
24  
25  
26  
27  
28  
29  
30

webbutveckling med öppna APIer, back-end systemintegration etc. Alla dessa funktioner stöds fullt ut av de andra versionerna förutom för Kentico CMS Ultimate där marknadsföringsfunktionaliteten fortfarande är begränsad, Kentico EMS och Corporate har funktion för alla delarna fullt ut utan tekniska begränsningar där Corporate tillåter oändligt med servrar vilket inte de övriga gör.<sup>31</sup> Kentico har funktioner för inbyggda bokningssystem, SEO, inbyggd WYSIWYG, Workflow, Mediabibliotek, Webanalyser, Kontakthantering, Nyhetsbrev samt E-handelfunktionalitet för exempelvis onlinebetalningar, kundvagn samt donationer för att nämna några delar av allt som CMS:et stödjer. Med andra ord är det ett väldigt stort och mångsidigt CMS med mycket funktionalitet för många kundgrupper.<sup>32</sup>

Kentico ger möjligheter att arbeta antingen via deras egna ”Portal Engine” som ger en möjligheten att skapa Kenticobaserade webbplatser direkt via ett webbaserat gränssnitt genom att använda färdiga ”Web Parts”-komponenter. Ingen riktig programmering krävs här, utan endast när man väljer att skapa sina egna komponenter till sin installation. Kentico erbjuder även möjligheten att skapa sina webbsidor eller webbapplikationer i Visual Studio med MVC eller med Web Forms.<sup>33</sup>

Kentico CMS's server kräver minst Windows 7 Service Pack 1, men har även stöd för 8 samt 10 & Windows Server 2008 R2 och uppåt. De rekommenderar även Microsoft .NET 4.5 alternativt 4.6, Microsoft IIS samt Microsoft SQL Server 2008 och uppåt.<sup>34</sup>

## 2.6 Mindbite SiteFactory

SiteFactory är Mindbite's egna CMS-system som de har utvecklat sedan starten av deras företag år 2000. Detta CMS utvecklas kontinuerligt och bygger på Microsoft .NET och C#. Som databas för CMS:et används Microsoft SQL Server.

Systemet har alla de standardfunktioner hos ett CMS för att hantera sidor, texter, artiklar, webbanalyser, SEO samt utbyggnadsmöjligheter i form av specifika moduler där man kan bygga upp valfri funktionalitet i C#. CMS:et har stöd för .NET Web Forms och i dagsläget finns inget stöd för exempelvis MVC eller Core som är nyare tekniker inom .NET.

Då CMS-systemet är utvecklat av dem själva finns inga licensavgifter eller liknande och systemet är inte open source.

---

31  
32  
33  
34

## 2.7 Agil Projektledning / Systemutveckling

För att ta de mest vanliga och använda metoderna att kunna jämföra Mindbite's projektlednings- och arbetssätt så valde jag att använda Scrum inom Agil Projektledning samt den traditionella vattenfallsmetoden i denna analys.

Det agila arbetssättet kan man beskriva som ett samlingsnamn för ett antal olika principer och metoder som i detta fall används i systemutveckling. Agil systemutveckling följer de grundtankar som formulerades i 'Manifesto for Agile Software Development'<sup>35</sup>.

I den agila systemutvecklingsmetoden har man som högsta prioritet att göra kunder nöjda genom tidiga leveranser och fortsatta delleveranser med fungerande och värdefull programvara som efterfrågats av kund. Värdefull programvara är det som uppfyller målen med projektet samt leverera den produkt som maximerar kundens nytta av det man fått gjort i projektet.<sup>36</sup> Man tillåter förändring under arbetets gång, vart man än befinner sig i arbetet ska inte spela någon roll utan programvaran sätts först och inte den kravspecifikation som specificerades i början av ett projekt för den omöjligt kan förutspå vilka förändringar och problem som kan uppstå under ett arbetes gång.<sup>37</sup>

Man ska leverera fungerande programvara kontinuerligt där man helst levererar med kortare mellanrum<sup>38</sup>, personer iblandade i ett projekt ska även kommunicera effektivt och dagligen för att undvika missförstånd och arbeta bra tillsammans.<sup>39</sup> Projekten ska byggas av motiverade individer och dessa ska man ge det utrymme och den hjälp och stöd som krävs och genom dessa delar lita på att individen får det arbete som givits görs.<sup>40</sup> Den allra effektivaste sättet att kommunicera inom ett utvecklarteam är att genom riktiga konversationer meddela den information som behöver meddelas, istället för dokumentation eller andra digitala tjänster då det blir så mycket man missar, med teknikens framfart finns det många möjligheter gällande videomöten etcetera, men hur koncentrerad är man faktiskt vid ett sådant, jämfört om man skulle föra en konversation i person med någon? När det är flera personer iblandade i ett sådant möte kan det även bli förväxlingar mellan vem som säger vad, och därav tar det även mer tid tillsammans med att man mycket lättare blir distraherad vid ett sådant möte, på grund av så mycket andra saker. I ett vanligt möte kanske man inte hade tagit upp sin mobiltelefon och kollat sin mail exempelvis, men detta är något som mycket enklare skulle kunna hända vid ett digitalt möte, och distraktioner bidrar till missförstånd och tidsfördröjningar.<sup>41</sup>

---

35

36

37

38

39

40

41

Framgång mäts i fungerande programvara då de satta kraven inte är fasta utan går att förändra under arbetets gång, därav mäter man framgång i ett projekt genom delar som är klara och fungerar, vilket innebär att den utvecklade produkten ska vara testad och godkänt av kunden och inte bara gjord efter specifikation som inte blev optimal och ändå markerad ”klar”.<sup>42</sup>

Simpliciteten att maximera delen arbete som inte görs, är en teknik som är viktig i det agila arbetssättet, vad detta innebär är att man hela tiden ska försöka effektivisera och göra sin programvara så simpel som möjligt efter vad som krävs av den, utan att göra saker som inte har krävts eller istället för att utveckla smart kod skriva onödigt mycket dokumentation för att beskriva vad man gjort.<sup>43</sup>

I ett agilt projekt har man även hellre flera personer som kan lite om allt, istället för några som är specialister på ett specifikt område, genom detta kan man utnyttja fler personers idéer och tankar och på så sätt komma på smartare designar och i slutändan bättre slutresultat. Detta sker genom en grupp som delar sina kunskaper med varandra och arbetar tillsammans på ett effektivt sätt och lär sig utav sina misstag. Detta föredras framför att det är flera specialister som arbetar på sina egna saker, och när väl saker ska bindas samman så fungerar ingen tillsammans utan måste förändras, vilket än en gång kräver arbete som hade kunnat undvikas – Maximera den delen arbete som inte görs.<sup>44</sup>

Det är viktigt för projektteamet att hela tiden som grupp titta på vad som gjorts och vad som behöver göras framåt, detta är viktigt för att gruppen genom dessa möten ska diskutera idéer av hur saker kan göras och därigenom kommer teamet att utvecklas snabbare och även de enskilda individerna genom bollande av tankar och hur saker och ting kan göras och eventuellt förändras.<sup>45</sup>

---

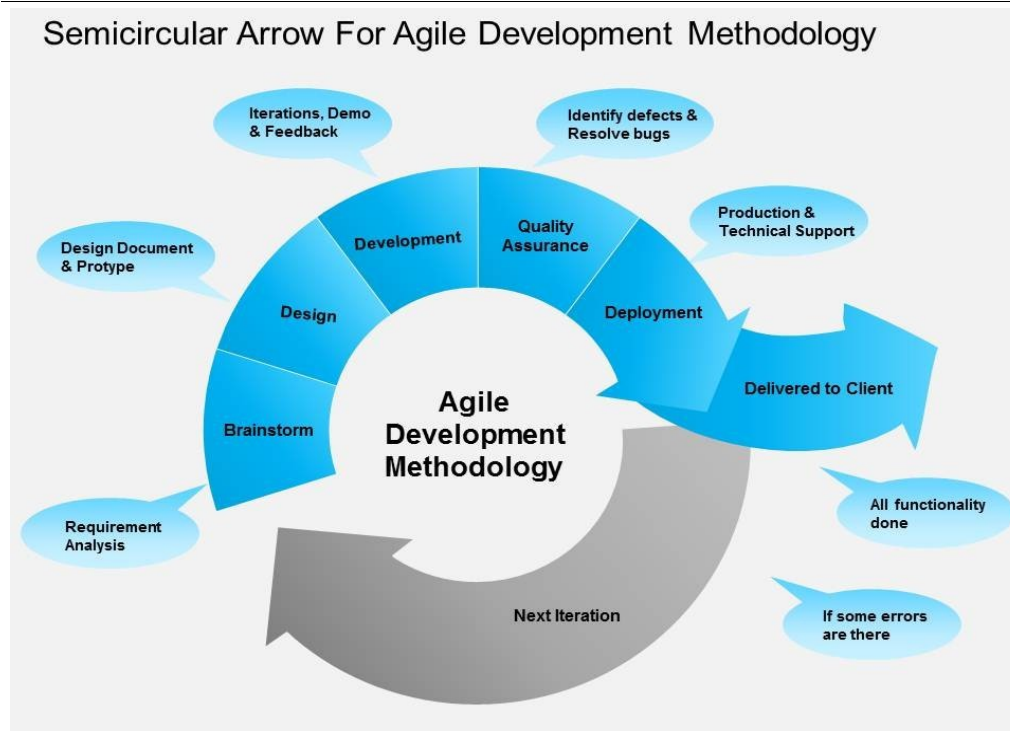
42

43

44

45





**Figur 2.1** – Beskrivning av den agila arbetsprocessen i en cirkulär process<sup>46</sup>

## 2.8 Scrum

Scrum är en agil systemutvecklingsmetod som bygger på de grundläggande delar som definieras i det agila arbetssättet som är den mest använda utav alla agila arbetsmetodiker. Scrum baseras på den agila utvecklingen där man under arbetsgången hela tiden kan och ofta ändrar lösningsalternativ och kravspecifikation allt eftersom arbetet fortgår. Vad som skiljer Scrum från övriga agila systemutvecklingsmetoder är vissa specifika koncept som man använder sig av i Scrum.<sup>47</sup>

I ett projekt där man tillämpar Scrum så har man nästan alltid tre roller<sup>48</sup>, man har en produktägare/ansvarig vars roll är att se till att visionen över den färdiga produkten förs över till utvecklingsteamet som är menat att utveckla den programvara som ska skapas. ”Product Owner” är även ansvarig för att från kunderna få så detaljerad information som möjligt för hur de vill att allt ska fungera och se ut och vilka övriga önskemål som finns, denne har även som ansvar att hålla kontakt med kunden med t.ex. möten och övrig kommunikation med själva teamet.<sup>49</sup> Denna person kan man säga är kundens röst till Scrum Teamet, som har mer kunskap bakom det tekniska och kan beskriva visionen på ett mycket bättre sätt en vad kunden hade kunnat göra direkt, här undviker man även missförstånd.

46

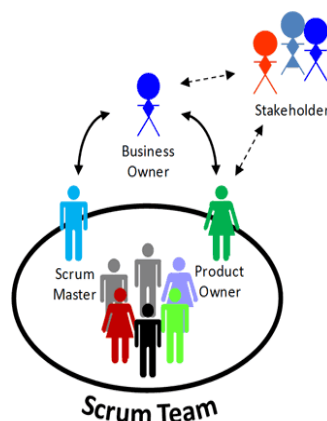
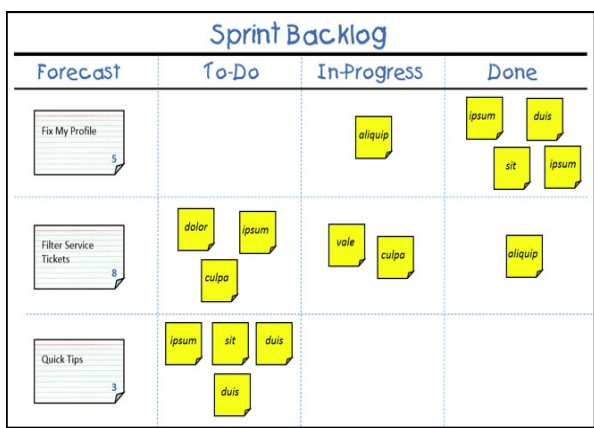
47

48

49

Utöver en ”Product Owner” så har man även ofta en Scrum Master. Vad denna person gör är att ta ledarrollen över teamet samt coacha och motivera teamet i deras arbete.<sup>50</sup> De största ansvarsområdena utöver detta för en Scrum Master är att även hjälpa produktägaren ta hand om produktens ”Backlog” och genom detta enkelt se till att inget arbete missas och förstås väl utav utvecklarna, tillsammans med detta ser även denna person till att teamet hela tiden kommer framåt i sitt arbete genom att driva på och motivera dem samt även se till att eventuella hinder i utvecklingens fortgång tas bort så att arbetet hela tiden kan fortsätta, vare sig det är ett externt eller internt hinder.<sup>51 52</sup>

Den sista rollen, som egentligen alltid är en hel grupp är själva utvecklingsteamet. Vart denna grupps ansvar ligger är ganska klart då detta är den grupp som bedriver själva utvecklingen av allt. Dess primära syfte är att utveckla och kontinuerligt leverera kvalitativ programvara i delleveranser till kund. Detta team uppgörs vanligtvis av 3-9 individer som gör allt arbete från analys till utveckling och slutliga tester och dokumentation och även teknisk kommunikation med kunder.<sup>53</sup> Dessa team ska vara självorganiserade och genom de flera medlemmarna täcka så mycket kunskap det går istället för att ha flera specialister som egentligen inte kan hjälpa varandra och utvecklas på samma sätt som ett team. Teamet ska följa en s.k. ”Sprint Backlog” som innehåller förutspåelser för framtida behov och ändringar, vad som ska göras, vad som håller på att göras samt vad som är färdigt, genom denna får man en enkel bild för vad som gjorts och vad som är närmast etcetera. I slutet av varje



sprint görs en delleverans.<sup>54</sup>

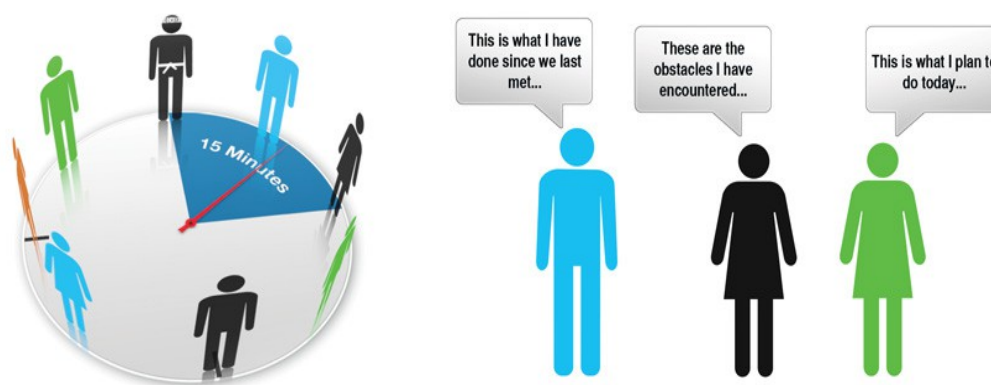
**Figur 2.2 | 2.3** – Sprint Backlog | Överblick över ett komplett Scrum Team<sup>55</sup>

56

50  
51  
52  
53  
54  
55  
56

I Scrum använder man sig utav sprinter. Sprinterna man använder bygger på tidsboxar för fasta tidsperioder, t.ex. en dag, vecka eller månad, men allra vanligast är tvåveckorssprinter. Här använder man sig utav den Sprint Backlog som skapas där alla aktiviteter och delar som ska göras definieras.<sup>57</sup>

I ett Scrumprojekt håller projektgruppen vanligtvis ett dagligt möte kallat ”Daily Scrum”. Dessa hålls oftast på starten utav dagen, då man då får hela resterande dagen till arbete och man kan även pricka av och sätta upp vad man som projektmedlem ska göra under dagen. Vanligtvis är dessa cirka 15 minuter långa, som en egen tidsbox som helst varken ska vara kortare eller längre för att upprätthålla diskussioner och göra mötet så effektivt som möjligt.



**Figur 2.4 | 2.5** – Illustration av 15 minutersmötet<sup>58</sup> | Viktiga delar<sup>59</sup>

Här får alla i projektgruppen säga sitt, och alla är tvungna att delta, man går var och en igenom vad man ska göra under dagen, vad man gjort dagen innan samt om man har några eventuella problem i sitt arbete, vare sig det gäller någon hårdvara som inte fungerar, eller om det är något som krånglar i själva utvecklingen.<sup>60</sup>

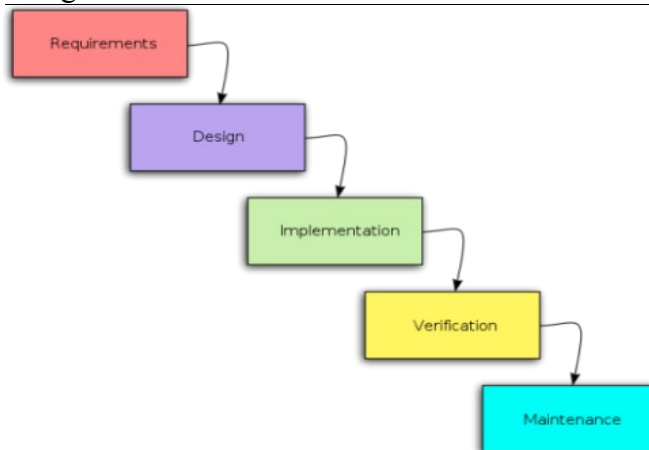
## 2.9 Vattenfallsmodellen

Vattenfallsmodellen är en annan typ av projektupplägg. Vattenfallsmodellen kan man säga kort och gott följer en process steg för steg där allt följer den plan som definierades i början av arbetet.

Processen flödar ständigt neråt i en stege genom alla faserna som ska genomföras, dessa ser i regel ut såhär:

---

57  
58  
59  
60



**Figur 2.6** – Vattenfallsmodellens faser <sup>61</sup>

I den initiala fasen, kravställningar och projektplaneringen så lägger man vanligtvis upp en tidsplan tillsammans med all funktionalitet som den slutgiltiga produkten är tänkt att se ut, här tar man oftast inte med hur allt ska fungera, bara vad som skall finnas med.

Det andra fasen i projektet vilket är designfasen så täcker man allt gällande applikationens design, det vill säga tekniska krav (val av programmeringsspråk, tjänster etc.). Här definierar man ofta all den information som krävs för att allt ska kunna implementeras i kod efteråt, hur allting är tänkt att byggas upp planeras, design av användargränssnitt och så vidare.

I det tredje och ett mycket viktigt steg som ofta tar den allra största tiden i ett projekt är själva implementationen av allt som man satt upp i kravställningen och ska helst följa de val gällande design som valts sedan det tidigare steget. Här skapas alla delar av utvecklarna, vanligtvis i olika delar som senare testas var och för sig, detta är för att se till att inga fel finns i dom enskilda delarna utan så allt att fungerar enligt specifikation.

Det näst sista steget innebär att man i första hand testar alla delar, för att sedan sätta ihop dem och därefter testa allt tillsammans, för att se att det fungerar korrekt. Här ska själva applikationen som helhet fungera som tänkt, alla tekniska delar som planerades att göras ska här vara gjorda och uppfylla de krav som satts ut. Slutligen i detta steg levererar man applikationen eller webbsidan till kunden och sätter upp denna på den domän eller server där den ska ligga om den är tänkt att t.ex. vara ett intranät eller dylikt.

Vad man sedan gör när själva utvecklingsarbetet av allt som planerades så krävs det ofta en del underhåll. Det förekommer ofta problem i kundens servermiljö eller domän som måste rättas till i form av patchar. Om kunden så önskar så utvecklas även uppdateringar och förbättringar av de delar som finns, men ibland utveckling av nya delar som kunden beställer till sin befintliga lösning.<sup>62</sup>

61

62

I vattenfallsmetoden har man mer traditionella roller i projektet, man har utvecklar likt den agila metoden som gör all kod till applikationerna som framställs och detta är i grunden deras enda ansvar tillsammans med att utveckla felfri kod då buggar kan sakta ner ett helt projekts arbetet. I denna metoden har man även alltid testare, detta arbetar främst med detta i de slutgiltiga stegen utav ett projekt där allt ska testas för sig, och sedan sättas samman där allt behöver testas igen för att säkerställa att allt funktion finns och så att applikationen uppfyller alla övriga tekniska och krav gällande design och användbarhet.

Projektet innehåller även alltid en projektledare som strukturerar upp projektgruppen med vem som ska göra vad baserat på medlemmarnas kunskaper. Denna individ ansvarar även för den slutgiltiga produkten och ska se till att inget lanseras innan det fungerar precis som tänkt och även inom den tidsram som satts.<sup>63</sup>

## 2.10 Fördelar och Nackdelar

Med **vattenfallsmetoden** finns det en del delar som talar för användning utav denna metod.

- Utvecklare och kunder enas i de inledande delarna utav ett projekt om precis vad som skall levereras i de första faserna, innan diverse uppdateringar och eventuella patchar. Detta gör projektplaneringen enkel och utvecklingen blir också enklare då man redan definierat hur allt ska byggas upp.
- Utöver tester och godkännanden samt olika möten så behövs ej kundens närvaro, då allt redan satts ut och bestäms i förväg.
- Det är enkelt att mäta hur långt man kommit i arbetet, då alla krav och design redan definierats i planeringsstadiet.
- All utveckling kan ske med mer säkerhet då man redan vet och kan läsa sig till hur allt ska fungera och se ut. Detta ger en mer förutsägbar design där man i de flesta fall undviker möjligheten att saker som kodats inte passar och agerar väl ihop.<sup>64</sup>
- Metoden är användbar där man redan vet hur allt ska se ut och hur allt ska fungera, alltså när man har ett väldigt klart mål gällande den slutgiltiga produkten.<sup>65</sup>

Det finns även en del nackdelar med denna metod, och många av dessa är anledningen till att denna metod börjat fasa ut i många fall.

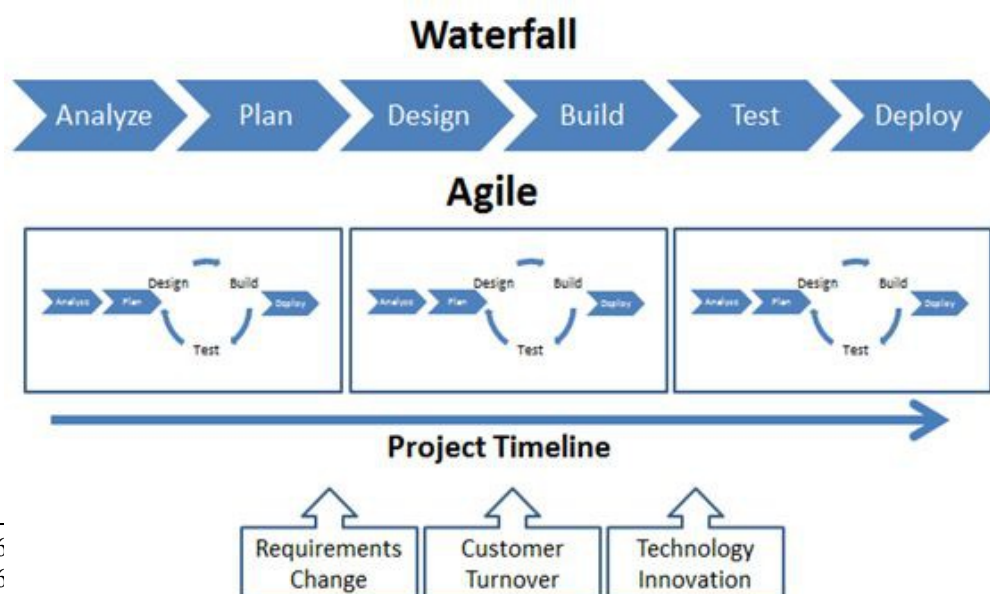
---

63

64

65

- En stor anledning är att i många fall är inte kunden speciellt tekniskt insatt gällande alla de olika delarna utav programmering, design etcetera och kanske inte riktigt förstår vad man allt betyder och när allt börjar sättas samman enligt dokumentationen som finns sedan de tidiga faserna så blir kunden missnöjd när denne ser vad den slutgiltiga produkten börjar se ut som. Vid denna tidpunkt blir det ofta mycket kostsamt, och även en teknisk utmaning att ändra en applikation, självklart beroende på storleken av det som behöver ändras.
- Kunden är sällan inblandad i processen mer än i de tidiga stegen, där de som tidigare nämnt sällan förstår alla detaljer och får svårt för att sätta ihop allt i deras huvud mer än eventuella wireframes och designskisser som kan visas upp. På grund av dessa kan det skapas missförstånd gällande slutresultat och problem som även dessa nämns i föregående punkt.<sup>66</sup>
- När man kommit så långt som testandet av allting där saker och ting börjar kopplas samman, kan det vara en svårighet att ändra något i applikationens kod.
- Ingen fullt fungerande programvara finns att testa förrn sent i utvecklingsprocessen, där det som tidigare nämnt vanligtvis är en teknisk svårighet att ändra på saker och ting.
- Modellen passar inte objektorienterade lösningar, då allt i sådana lösningar kopplas samman direkt i utvecklingsarbetet och inte vid testandet för lätthet och undvikandet av framtida problem.<sup>67</sup>
- Fler projekt misslyckas än projekt som använder sig utan den agila projektmetodiken.<sup>68</sup>



**Figur 2.7** – Vattenfallsmodellens upplägg jämfört med den agila modellen <sup>69</sup>

Den **agila projektmetodiken** har även den självklart både fördelar och nackdelar:

- En stor fördel med denna typ av arbetsmetod är att kunden i flera steg har möjlighet att se hur applikationen eller webbplatsen ser ut för tillfället och testa den funktion och design som finns för tillfället, detta då projektet här hela tiden görs i små delleveranser efter varje sprint där ett visst antal delar implementeras och testas.
- Om det är viktigare att producera en applikation eller webbsida snabbt för kunden ska kunna nå ut till sin marknad snabbt så kan man enkelt med denna arbetsmetodik producera en basversion utav den fulla versionen som i slutändan ska byggas.<sup>70</sup>
- Nära samarbete mellan kund och projektgruppen ger ett bättre resultat som kunden hela tiden får vara med och påverka där ändringar kan göras under hela projektets gång. Detta då fungerande programvara levereras med regelbundna kortare mellanrum där veckor är vanligare än månader som vattenfallsmetoden ofta innebär.
- Den agila projektmetodiken har en högre procentuell del där arbeten slutfördes med bra resultat och väldigt låga procent som misslyckats.<sup>71</sup>

Det finns även en del nackdelar, som talar för en annan alternativ arbetsmetodik:

- I vissa fall, främst i större projekt så kan det vara svårt för projektledningen att utvärdera hur mycket arbete som faktiskt kommer att krävas utav det hela, då krav hela tiden kan komma att förändras.
- Kopplat till tidigare punkt, projektet kan väldigt lätt dra iväg bortom vad som kan ha tänkts i början av projektet, både gällande tid och omfattning, detta om kunden eventuellt är oklar med vilken färdig produkt denne vill ha och det blir mycket ändringar etc.
- Det finns inte samma utrymme för mindre erfarna programmerare att delta i projekten, då mycket egna val ska tas. Det fungerar i samband med mer erfarna utvecklare, men detta är något som inte hade krävts på samma sätt i vattenfallsmetoden där hur allt ska byggas upp och fungera redan finns definierat.<sup>72</sup>

---

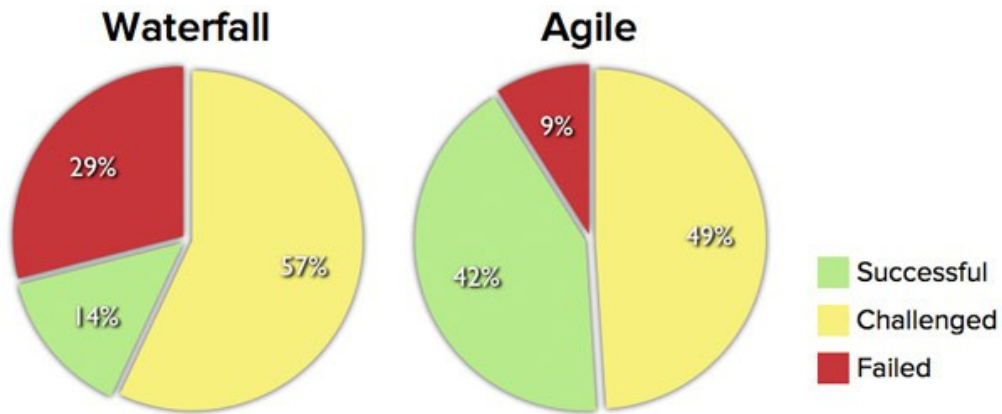
69

70

71

72

- Det finns större krav på att alla medlemmar ska arbeta från samma fysiska plats då detta är enklare att hantera i ett projekt, detta är inte alltid möjligt och det är relativt vanligt att utvecklare arbetar från en annan plats idag.<sup>73</sup>



Source: The CHAOS Manifesto, The Standish Group, 2012.

**Figur 2.8** – Agila metoden jämförd med vattenfallsmetoden

---

73



## **3 Metod**

### **3.1 Övergripande metod**

För att genomföra denna undersökning kommer jag att genomföra följande steg:

- 1) Sätta upp de kriterier som Mindbite kräver hos ett CMS.
- 2) Med verktyget Csmatrix samt insamlad fakta undersöka vilket/vilka CMS passar kraven och enkelt ställa upp detta i en samlad jämförelse.
- 3) Genom intervjuer undersöka utvecklings- och framtidsmöjligheter med SiteFactory CMS.
- 4) Utifrån insamlad bakgrundsinformation, jämförelser, praktik och intervjuer utvärdera samt ge förslag och motiveringar till framtida CMS och övriga system.
- 5) Analysera företagets arbetsprocess, vilka delar de använder sig av i projektmetodiken.
- 6) Utifrån samlad bakgrundsfakta och praktik utvärdera Mindbite's arbetsprocess. Föreslå och motivera förändringar för att optimera arbetsprocessen.
- 7) Vidare analysera resultaten och varför jag fick de resultat jag fick. Även slutsatser generellt kring arbetet, och även ta upp eventuella rekommendationer till fortsatt arbete.

### **3.2 Kriterier hos CMS**

I det allra första steget kommer frågor att ställas mot företagets teknik- och utvecklingschef för att få upp en bra bild över vad företaget i fråga och dess kunder kan begära hos ett CMS.

### **3.3 Undersöka passande CMS:er**

Här kommer en litterär studie att genomföras med insamling av bakgrundsfakta till samtliga valda CMS:er i denna undersökning. De valda systemen kommer att jämföras genom webbverktyget Csmatrix som enkelt jämföra specifikationer av olika CMS:er som finns tillgängliga.

### **3.4 Utvecklingsmöjligheter**

Utifrån frågor till företagets ledande utvecklare kommer jag överse vilka framtida möjligheter som finns med att utveckla deras befintliga CMS mot nyare tekniker.

### **3.5 Utvärdera & föreslå CMS och övriga system**

Till grund för detta kommer jag att använda den bakgrundsfakta som samlats till uppsatsen och genom denna fakta tillsammans med de frågor och resultat från Cmsmatrix samt intervjuer kommer jag att utvärdera deras egna CMS, och se gentemot andra CMS:er ifall det är värt att utveckla vidare, eller ersätta deras nuvarande mot ett annat. Samt eventuellt ifall det finns behov av övriga hjälpsystem för att förbättra arbetstakten hos verksamheten.

### **3.6 Samla fakta kring arbetsprocessen**

Genom deltagande i företagets arbetsprocess, i de steg som genomförs från planering till slutgiltig produkt kommer jag att notera de steg som används, hur man använder sig av möten, planering, arbetet med själva utvecklingen och hur deras arbetsgrupp ser ut.

### **3.7 Utvärdera arbetsprocess**

Genom denna fakta och genom det jag noterat under min praktik kommer jag att genomföra en undersökning gentemot de två projektmetoderna jag valt ut som de allra vanligaste och se över hur de matchar mot dessa, och hur de hade kunnat ändra sin arbetsprocess för att förbättra deras resultat och tidsanvändande. Till detta kommer även föreslag och motiveringar komma till eventuella förbättring hos företaget i deras sätt att arbeta.

### **3.8 Analys & slutsatser**

Jag kommer att analysera de resultat jag fått och knyta an och gentemot de frågeställningar och mål jag satte upp med arbetet, vad jag ville få fram, hur det har gått, varför jag fick de resultat jag fick etc. Även föreslå eventuellt fortsatt arbete och vad man vidare hade kunnat göra för att bedriva vidare liknande undersökning hos företaget.

### **3.9 Utvärderingskriterier**

Genom frågor mot företagets ledande utvecklare har jag kommit fram till dessa kriterier som ställs hos ett CMS för att det ska användas hos dom.

- **Grundfunktionaliteten i ett CMS**

Hur man kan strukturera upp/organisera sidor, systemets användbarhet, roller, samt övrig standard CMS-funktionalitet som att radera, lägga till, & uppdatera sidor.

- **Licens**

CMS:et ska för att kunna användas i en större utsträckning bygga på en gratislicens. Här det bästa valet MIT-licensen för Mindbite som ger en möjligheten att använda systemet kommersiellt, distribueras vidare, modifieras samt även användas privat.<sup>74</sup>

- **Support**

Vid användande av ett icke egenutvecklat CMS är det viktigt att detta CMS har ett bra community samt lämplig support ifall problem uppstår med kundens produkt. Detta kan vara antingen support via exempelvis e-post eller ett bra forum med mycket hjälp.

- **Möjlighet att bygga helt egna sidtyper**

Att kunna utveckla helt egna sidtyper utan någon autogenererad kod följer med är ett kriterie för att kunna vara flexibla och göra kundspecifika lösningar.

- **Versionshantering av artiklar och sidor**

Man ska enkelt genom CMS:et kunna gå tillbaka till en gammal version av en specifik sida eller artikel och återställa den till som den var tidigare.

- **Stöd för flera språk**

Med detta menas stöd för att ha flera språk tillgängliga i systemets gränssnitt. Det ska t.ex. finnas möjlighet att ändra mellan Svenska och Engelska.

- **Webbplatsanalys/statistik**

- **Byggt i Microsoft .NET-miljö**

CMS:et måste vara byggt i Microsoft .NET och bygga på programmeringsspråket C#.

För att få fram rättvisa resultat mot arbetsprocessen kommer jag undersöka metodiken på Mindbite mot övriga projektmetodiker enligt följande kriterier.

- 1) **Planering, hur planeras arbetet gentemot övriga tekniker? Sätter man upp fasta krav?**

- 2) **Används mycket dokumentation?**

---

74

- 3) **Hur genomförs arbetet, vem gör vad? Möten? Delleveranser? Kontakt mellan utvecklare?**
- 4) **Förändringar under arbetets gång?**
- 5) **Används s.k sprinter eller genomförs arbetet från början till slut utan stopp?**
- 6) **Arbetar alla från samma plats eller finns det medarbetare på andra orter?**
- 7) **När sker leveranser?**

## 4 Resultat

De grundläggande kraven för att forma undersökningen av ett passande och möjlig ersättare till Mindbite's nuvarande CMS – SiteFactory sattes upp.

### 4.1 SiteFactory funktionalitet

Företaget Mindbite AB startades år 2000 och har nu funnits i branschen i 17 år där de sedan start arbetat med utveckling av webbplatser och diverse intranät och program. Då Mindbite startade fanns det inte många CMS som passade in riktigt till deras profil och önskemål, och därav började istället att börja utvecklingen av ett eget, SiteFactory 1.0. En övervägande del till att de väljer att fortsätta med deras egenutvecklade CMS är även att de alltid kan specialanpassa lösningar efter kundens behov där man annars hade behövt förlita sig på en underleverantör som utvecklar CMS:et.

Genom mina analyser och utvecklingsarbete har jag gått igenom SiteFactory och fått en god bild över vad CMS:et erbjuder för sorts funktionalitet. När man först kommer in i systemets startsida och gränssnitt ser man får man direkt en bra överblick över all den funktionalitet som finns och kan enkelt navigera sig vidare i systemet.

I CMS:et finns det funktioner inbyggda som hela tiden varnar för eventuella fel i webbplatsen, och även verktyg kopplade till Google Analytics och SEO. Där man kan se eventuella fel och förbättringar som kan göras på webbplatsen kallas ”Webbplatsanalys” och är en egen flik i systemets gränssnitt. Här varnas det för mindre saker som t.ex. att det var länge sedan sidan uppdaterades, om sidan är sökmotoroptimerad eller ej, samt t.ex. att en specifik sida saknar en titel eller liknande. Man kan genomföra dessa analyser själv, eller så genomförs regelbundna analyser av webbsidan automatiskt.

Man kan även i systemet registrera nya användare direkt från CMS:et, här kan man välja olika roller som t.ex. Administratörer eller Author som har rättigheter att skriva nya artiklar och även ändra och radera dessa. Administratörer har utökade befogenheter som exempelvis skapande av nya användare och sidor.

Via CMS:et kan man även i vanlig ordning skapa nya sidor, undersidor samt även välja sidtyper på de sidor man skapar. Sidtyperna finns lagrade i servern och man kan enkelt skapa nya i Visual Studio där man kan bygga upp dem hur man vill. Systemet har även en flik för filhantering, här kan man hantera uppladdade bilder och filer som synkas till CMS:et via servern. Man kan även direkt via CMS-systemet ladda upp filer och skapa nya mappar.

Själva CMS:et har som grund endast funktionalitet för att skapa nya sidor och fylla dessa med information, skapa nya användare, webbplatsanalyser & SEO

samt hantering av filer och skapande av nya bloggposter om man valt att skapa en sida med sidtypen CmsBlog. För att utöka funktionalitet i systemet så bygger detta på "Moduler". Varje modul har en egen funktion, och vanligtvis en kategori. Dessa är kopplade mot Microsoft SQL Server och vid klick på en modul kommer man till en samlings sida där man ser alla poster uppräddade i en lista. I dessa listor kan man ändra, radera samt skapa nya poster.

För att ta ett exempel så skapade Mindbite en sida byggd på SiteFactory åt UKM (*Ung Kultur Möts*). Denna verksamhet arrangerar festivaler över hela Sverige och behövde funktionalitet för att skapa nya städer, län och även länder. Dessa delar skapades som moduler för att fyllas med olika länder som deltagare kan komma ifrån, städer där festivaler arrangeras samt även kommuner där festivaler ska arrangeras. Dessa sköts endast i CMS:et och kan inte förändras från själva webbplatsens gränssnitt. Dessa moduler kan sedan koppas mot andra delar på sedan genom s.k. Foreign Keys där de ärver värden från varandra, varje kommun ärver t.ex. ett län och ett land som måste definieras vid skapandet av en ny kommun.

Det skapades även en modul för konton, som även är kopplat mot själva webbplatsens gränssnitt. Anledningen till att det finns en specifik modul för konton var för den anledningen att UKM's anställda ska kunna godkänna konton och se till att all information är korrekt, dessa görs alltså manuellt via CMS:et.

Varje användare får här ett autogenerated UKM-ID som sedan används för att koppla en användares registrering mot en festival. Festivalerna sköts på liknande sätt och man skapar nya via den skapade modulen i CMS:et. Dessa listas sedan ut på webbplatsen efter datum, kommun, och även i en lista vid registrering till festival för en användare. Alla dessa moduler skapas i Visual Studio och skrivs i C#. Detta är extra funktionalitet som kunder får betala för att det ska skapas och är svårare för dem att skraddarsy ett CMS som inte är deras eget, enligt dem själva.

Detta är den funktionalitet som deras eget CMS har och även hur det är tänkt att användas. Genom dessa moduler som Mindbite skapar vid speciella önskemål är även viktiga för deras möjlighet att skraddarsy lösningar för företag som har speciella behov i deras innehållshanteringssystem.

## 4.2 CMS

Jag fick fram följande resultat från verktyget där Kentico, DNN och Umbraco jämfördes utifrån deras specifikationer enbart.

Från jämförelsen i bilaga 1-6 får man resultat för vad för sorts funktionalitet som de olika systemen behöver, här har jag endast tagit ut de delar som innehåller krav som Mindbite ställer hos ett CMS som de ska använda. För att ta fram det viktigaste för att eventuellt skildra fram ett av dessa system som är det bättre så genomgicks kriterierna som sattes upp i tidigare avsnitt:

- **Grundfunktionaliteten i ett CMS**
- **Licens**
- **Support**
- **Möjlighet att bygga helt egna sidtyper**
- **Versionshantering av artiklar och sidor**
- **Stöd för flera språk**
- **Webbplatsanalys/statistik**
- **Byggt i Microsoft .NET-miljö**

Alla av de utvalda CMSerna innehåller den självklara funktionaliteten som ett CMS egentligen bygger på, möjlighet att kunna skapa, redigera samt ta bort sidor och inlägg. Samtliga CMSer ger även möjligheter till att skapa bl.a. undersidor och tillåter även användande av teman.

De tre innehållshanteringssystemen bygger enligt Mindbites krav även på .NET plattformen och använder IIS som server samt MSSQL som databas, det går dock att i Umbraco att även använda sig utav MySQL som databasalternativ. Samtliga CMSer är även gratis att använda även att Kentico inte är ett open-source CMS utan har vissa begränsningar gällande användandet och vad för sorts kapacitet man har i systemet.

Det finns t.ex. begränsningar gällande användare och filer som skapas utan en betalversion utav detta system. Man måste även ta med deras logotyp på webbplatsen man skapar för att man ska få använda systemet gratis. Man får heller ingen support utöver deras communityforum som finns tillgängligt för alla. För att få tillgång till support via telefon och e-post krävs betalversioner utav Kentico.

Forum är en bra källa för att få hjälp, dock är här Kenticos användarbas är mycket mindre än de övriga två med endast 25000 aktiva webbplatser, och därav anses denna support som otillräcklig jämfört med de andra två med ett stort antal fler användare då det kan ta tid att få svar på eventuella frågor man som utvecklare kan ha.

Enbart dessa punkter gör Kentico opassande för Mindbite och utesluts därav från närmare analys då det inte passar några av de allra viktigaste kraven som företaget vill ha hos deras CMS, som t.ex. gratislicens med fritt användande,

open-source med tillgång till källkod, support och utöver dessa delar även webbstatistikanalys och flerspråksstöd.

För att gå vidare med undersökningen jämfördes därav endast Umbraco samt DNN Platform på de punkter som jag har fått fram som kriterier. Både Umbraco och DNN är open-source och båda är licenserade under MIT-licensen vilket enligt företaget är den bästa licensen för deras utveckling och även för deras kunder då det erbjuder dem att erbjuda ett gratis CMS.

De båda erbjuder även funktioner för att kunna hantera versioner utav innehållet på den skapade webbplatsen i det tillhörande CMS, vilket även är en viktig punkt för att ge ett extra lager säkerhet för deras kunder och möjligheter att enkelt återhämta tidigare versioner av sidor och artiklar. Inget utav dessa två CMS ger någon gratis utökad support via t.ex. chat, mail eller telefon utan detta är sådant som kan köpas till för en extra kostnad. De två systemen har dock stora communityforum med ett stort antal användare på grund av antalet registrerade sidor med dessa system (ca 400.000 respektive 750.000). Detta ger en som användare stora möjligheter att få hjälp med sin utveckling. I Umbraco får man tillgång till umbraco.tv där det finns ett stort antal videor och beskrivningar på hur man använder systemet och hur man utvecklar sidor inom det.<sup>75</sup> Detta finns liknande inom DNN i deras video library.<sup>76</sup>

De båda innehållshanteringssystemen har även support för att skapa flerspråkiga sidor vilket även är en viktig punkt för att Mindbite även ska kunna inrikta sig mot både internationella kunder såväl som svenska kunder som vill utöka sin kundbas genom flerspråkssidor. Umbraco och DNN stödjer har även inbyggd funktionalitet för analys av webbplatsstatistik likt SiteFactory.

DNN har funktionalitet för att skapa nya sidtyper direkt via deras UI, detta skapar en ny sidtyp/template med viss färdiggenererad content på dem, detta är oönskat, det finns dock utöver detta utvecklingsmöjligheter att skapa helt egna sidtyper utan någon autogenererad kod på dem.<sup>77 78</sup>

Detta är även möjlig med Umbraco, dock kan man här skapa ett template direkt via systemets UI och även redigera koden här för att väldigt enkelt få en helt tom sidtyp att bygga upp hur man vill.<sup>79</sup> Detta är något krångligare i DNN där man istället måste gå in i filstrukturen för att skapa en ny template och sedan redigera denna i exempelvis Visual Studio.

Då båda dessa system erbjuder all den funktionalitet så skedde fler intervjuer med företagets teknik- och utvecklingschef kring dessa två för att se om de hade testat något av dem och vilket de hade fördragit och varför.

---

75

76

77

78

79



Robin Månsson på Mindbite svarar här att de under 17 år i branschen testat flera CMSer, däribland både DNN och Umbraco. Anledningen till att de föredrog Umbraco av dessa två var att Umbraco innehåller mer utav allt de nödvändigaste som de kräver i sitt grundutförande, utan att behöva ladda ner diverse tillägg för att det ska fungera som de vill ha det anpassat till deras verksamhet. Exempelvis att man väldigt enkelt får konfigurera alla sidtyper man skapar från ett blankt papper, detta kan inte göras på samma sätt i DNN, detta ger en större flexibilitet och Umbraco som helhet anses även utav utvecklarna som ett mer flexibelt och användbart CMS än DotNetNuke.

Då utvecklarna på företaget anser Umbraco som det bättre CMS:et utav DNN och detta så har jag genom mina intervjuer även fått reda på att Mindbite redan byggt en kundlösning i Umbraco. Detta gjordes på grund av att de ville kunna erbjuda deras kunder flera alternativ gällande CMS:er beroende på vad för slags funktionalitet som behöver byggas upp i det projekt som gäller.

Hela utvecklarstyrkan har därmed testat och utvärderat Umbraco för att lära sig mer kring övriga CMS:er på marknaden och även se vad övriga system har att erbjuda utöver deras egna. På grund av den fakta och den information jag fått reda på utifrån mina intervjuer och praktik så ser jag Umbraco som den enda möjliga lösningar som uppfyller alla Mindbite's krav och är även något som de har tidigare erfarenhet utav, något de inte har lika mycket utav i DNN även att de båda erbjuder liknande funktionalitet gällande de delar som Mindbite önskar hos deras innehållshanteringssystem.

Den uppenbara nackdelen med SiteFactory är att det bygger på teknik som börjar fasas ut inom .NET-utveckling vilket är .NET WebForms. Det är inte vanligt att nya webbplatser och system längre utvecklas i WebForms på grund av många anledningar, däribland snabbhet och den renare koden man får vid utveckling i nyare tekniker som MVC och Core.

För att SiteFactory ska vara ett konkurrenskraftigt alternativ kommer förändringar behöva ske med denna produkt, där CMS-systemet antingen behöver vidareutvecklas mot nyare teknik, eller möjligtvis användas endast som back-end system som i sin tur kan kopplas mot diverse sidor som bygger på helt andra tekniker via WebAPI. Detta ger Mindbite stor flexibilitet gällande vilka kunder de kan leverera lösningar till och på vilka sätt dessa lösningar kan skapas.

Ett exempel kan vara att koppla Mindbite SiteFactory mot front-end på en Wordpress-webbplats för att genom deras egna tekniker och anpassningsmöjligheter utveckla lösningar som hade varit krångligare att framställa i exempelvis Wordpress men ändå leverera dessa till en helt annat webbplatsteknik.

Denna fråga ställdes även till företagets chefsutvecklare Robin, han svarar att det är en fråga de ställer sig ofta, och det är detta som precis tagits upp som funderingarna går mot där det endast servrar data mot valfritt annat CMS i

front-end. Han nämner även hur de samtidigt jobbar för att modernisera SiteFactory något för att kunna blanda Web Forms och MVC i samma installation.

Det påpekas även att det kostar väldigt mycket pengar att utveckla och underhålla SiteFactory, men han ser det som en stor fördel att de har full kontroll på deras egen produkt där dem kan göra snabba rättningar om fel uppstår, men även en stor flexibilitet gällande precis hur de vill att allt i systemet ska fungera vilket ger dem en stor fördel till att skraddarsy lösningar då de kan bygga in mycket funktionalitet direkt i själva CMS-systemet.

Utifrån dessa svar och all den fakta som samlats in så ser resultatet i denna fråga ut som så att Mindbite bör behålla SiteFactory, och bygga vidare på det då det ger dem som företag så många fördelar, både för företaget samt deras kunder som de kan skraddarsy lösningar åt på ett annat sätt. Utöver att behålla SiteFactory bör de även ta vara på de möjligheter som finns med att integrera deras egna CMS och serva data till ett annat CMS, precis som företaget även har testat en del.

Detta ger fördelar som att de kan expandera en befintlig webbplats, vare sig den bygger på samma CMS eller inte med speciell funktionalitet som inte det befintliga CMS:et eller statiska webbplatsen har. I det fall kommer CMS-systemet endast agera back-end för datahantering via deras speciellt framställda ”moduler” för syftet som de är tänkta att uppfylla, detta blir en mer skraddarsydd lösning en vad som hade kunnat uppnås i många andra befintliga CMS:er på marknaden.

Det kostar som tidigare nämnt mycket pengar att underhålla och utveckla ett CMS, men det ger dem mycket fördelar som vägs upp av nackdelarna. Även att de bibehåller deras utveckling i och av deras eget CMS-system gör heller inte att möjligheterna med att utveckla webbplatser i Umbraco försvinner utan detta är något som fortfarande bör utnyttjas för mindre specialanpassade system som skall byggas samt för kunder som eventuellt önskar ett annat CMS. Då passar Umbraco på grund av deras tidigare erfarenhet av utveckling i Umbraco.

### **4.3 Arbetsmetodik**

Genom att delta i Mindbite's arbetsprocess där frågor mot företagets anställda tillsammans med anteckningar som förts under examensarbetets gång så har jag genom detta arbete fått upp en bra bild av hur webbyrån arbetar med de olika stegen och hur de lägger upp planeringen för ett projekt.

Då jag genom arbetet har fått se hur Mindbite arbetar från planeringsstadiet och själva utvecklingen till färdig produkt och lansering. Innan projektet sätter igång bedriver Mindbite en undersökning om verksamheten projektet gäller då

de vill få en bra bild över det företag det gäller. Här tar man fram de mål man har, syfte, möjligheter, utmaningar, struktur samt funktion och målgrupper.

Utifrån denna planering som gjorts tar man fram en översiktlig tidsplan och definierar de mål som är tänkta att uppnå. Här beskrivs saker och ting som innehåll, funktionalitet samt design och informationsstruktur.

Utifrån företagets grafiska profil, från t.ex. logga eller tidigare webbplats så tas det fram designskisser och wireframes. I de flesta fall används även storyboards för att få en bild över hur allt är tänkt att fungera praktiskt sett. Vid godkännande av börjar deras arbete. De använder i de flesta fall deras eget CMS SiteFactory, men kan även komma att använda t.ex. Umbraco beroende på kundens önskemål. Här börjar man med back-end utveckling utav den sida eller applikation som ska byggas. Man programmerar all den bakomliggande funktionalitet till den webbplats som ska byggas där man bygger moduler i SiteFactory för informationshanteringen i CMS:et.

När utvecklingen back-end är klar startar front-end utvecklingen där all informationshantering som byggts upp av programmerarna kopplas mot front-end delen av webbplatsen där allt syns mot användaren. Här kopplas sidorna upp mot databasen för att fungera med deras CMS och sidmallar och den övriga designen skapas, om detta inte redan gjorts i mindre utsträckning för t.ex. menyer etc. I produktionen görs även sökmotoroptimering och eventuellt koppling mot sociala medier samt responsiv design.

Sedan ska det framställda systemet installeras, testas samt driftsättas. I detta steg genomförs alltid även en kortare utbildning med kunden så de förstår hur dom ska använda CMS-systemet på ett fullgott sätt och förstår den funktion som byggts upp. Genomgångar av webbplatsens innehåll och funktionalitet, om allt godkänns publiceras webbplatsen. När webbplatsen väl är driftsatt så jobbar de aktivt med att vidareutveckla sidan efter kundens önskemål med eventuella förbättringar som kan komma.

Mindbite arbetar med korta cykler där de hela tiden avstämmer med kunden och arbetar med delleveranser av olika delar av webbplatsen. Själva grundwebbplatsen färdig för driftsättning sätts upp till en början, sedan sker delleveranser med de olika delarna av webbplatsen allt eftersom dessa cykler pågår. De har fasta möten varje måndag, men utvecklarna diskuterar även lösningar vid de pauser som finns under arbetsdagen så alla vet vad dem ska göra och man kan bolla idéer. De vill alltid att kunden ska vara involverad i projektet för att hela tiden kunna anpassa sig efter kundens behov. Detta fungerar bra då de sällan spikar en projektplan till 100% utan förändringar är välkomna under arbetets gång. Genom denna process kan Mindbite leverera i tid och inom budget.

Utifrån dessa delar kan man tydligt se att inom de flesta kriterierna så arbetar Mindbite mot en agil arbetsmetodik. Några punkter som utmärker sig speciellt är exempelvis deras sätt att använda delleveranser och "sprinter" där de har

korta arbetscykler där de har vissa saker som ska göras, dessa håller ofta på ungefär 2 veckor. De använder sig även utav en sprintlogg liknande det i **Figur 2.2** där det som skall göras finns uppsatt och de även bokför timmar på varje del samt totalen, detta kan ses i [Bilaga 7](#).

En annan punkt som utmärker sig är att företaget är öppna för förändringar längs med arbetets gång, så länge det är OK för båda parter och inte projektplanen är helt spikad från början, detta varierar från projekt till projekt beroende på vilken kund det gäller etc. Detta är en punkt jag anser man skulle kunna arbeta på, detta av anledningen att alltid försöka jobba så agilt som möjligt, om nu valt att arbeta med den agila projektmetodiken då det kan bli rörigt att blanda exempelvis vattenfallsmetoden och det agila arbetssättet.

För att arbeta mer mot den agila metodiken skulle företaget även kunna ha fasta, korta och koncisa möten varje dag på exempelvis 10 till 20 minuter där de går igenom allt för dagen istället för att endast ha ett fast möte på måndagar. Gruppen kommunicerar väl men det kan ändå uppstå missförstånd av vad som kan och ska göras. Dessa fasta möten skulle till stor del utplåna dessa problem och istället ge Mindbite bättre möjligheter att få det som ska göras gjort effektivare.

Utvecklingsteamets kommunikation är på en bra nivå och de delar kunskaper på ett bra sätt där alla arbetar tillsammans för att skapa bra produkter med deras arbete. En anställd arbetar remote och detta är något som fungerar väl med anledningen att de använder sig till största del utav agil projektledning. Denna utvecklare deltar i det mötet som hålls på Måndagar med hjälp utav Skype, men hade kunnat vara mer delaktig i företaget om dagliga korta möten hölls istället för det enskilda veckomötet. Denna person kan ibland ha missat saker och svårt att veta vad denne ska göra, detta löses dock fort via t.ex. telefon men hade helt kunnat undvikas med ännu en gång: dagliga möten.

Vad Mindbite saknar är även fasta roller som man kan ha i ett scrum-team i agil systemutveckling. Det som kan ses här är att även att de inte har några fasta utnämnda roller i verksamheten så kan man ändå säga att de arbetar mot sina roller, även att det inte är utsatt vem som är vad. Företaget hade kunnat använda sig utav fasta roller som exempelvis en Scrum Master, men det tror jag inte hade spelat någon roll för deras framgång. Chefen och projektledaren på företaget har hand om t.ex. deras Sprint Backlog, kommer med ideér, motiverar projektteamet och planerar arbetet. I sin tur gör utvecklarna dokumentation i den lilla mån som görs, men fokuserar främst på att producera högkvalitativ kod och bra produkter. Kommunikation till deras "Product Owner" som man kan kalla det, sker även detta via deras Chef/Projektledare som sköter den mesta kommunikation utåt mot deras befintliga kunder.

I övrigt syns det inga problem med verksamhetens sätt att arbeta och lägga upp projekt utan här sker ett väldigt professionellt arbete med professionella medarbetare med hög kunskapsnivå och erfarenhet. Chef, sälj och projektledare har bra kontakt med utvecklarna och har även genom tidigare erfarenhet utav

kodning en bra förståelse för koden som utvecklarna gör och kan därav vara mer insatt i arbetet.

#### 4.4 Slutsatser från utvärdering av CMS

Det som jag fått bekräftat genom mina utvärderingar och specifikationsjämförelser utav CM-system samt intervjuer och praktik fått reda på är följande:

- Företagets egenutvecklade CM-system gör sig bra i deras verksamhet på grund av full kontroll på produkten och möjlighet att skraddarsy lösningar
- SiteFactory kan och bör kompletteras med andra CMSer som inte kräver lika skraddarsydda lösningar utan kan byggas på en befintlig plattform med nyare tekniker
- SiteFactory bör vidareutvecklas mot nyare tekniker stegvis för att göra systemet mer konkurrenskraftigt, alternativt användas som back end och endast servra data mot befintlig webbsida som inte bygger på SiteFactory CMS.
- Fördelar med att behålla deras egna CMS är att de har full kontroll på produkten, men nackdelar blir att det måste vidareutvecklas mot nyare tekniker och detta blir kostsamt. Att istället använda ett eget helt ger dem ett kostnadseffektivt alternativ dock inte lika stor flexibilitet.

#### 4.5 Slutsatser från utvärdering och analys av projektmetodik

Genom de analyser som genomfördes utifrån intervjuer samt delvis deltagande i deras arbete har jag fått en bra bild över hur saker och ting fungerar på företaget samt hur de lägger upp deras arbete med nya projekt.

- Mindbite använder sig till största del av agil projektledning, inte nödvändigtvis Scrum till punkt och pricka då de saknar de specifika rollerna – iallafall på pappret. De uppfyller dock övriga kriterier för Scrum med några undantag.
- De bör se över deras mötesstrategi och istället ha korta och koncisa möten fast varje dag istället för en gång i veckan för att alla ska ha en bra bild över vad de ska göra.
- Företaget fokuserar främst på det agila arbetssättet och har inga delar som utmärker dem för att använda traditionell vattenfallsmetod.

## 4.6 Rekommenderade åtgärder

Jag har kommit fram till följande rekommenderade åtgärder och framtidsplaner för företaget:

- SiteFactory bör fortsätta att användas, men under en längre tid sakta vidareutvecklas mot nyare tekniker på grund av den tid det kräver.
- Istället för att behålla eller byta ut det ena eller andra komplettera deras egna CMS med ett annat, i detta fall passar Umbraco bäst under tiden som deras egna vidareutvecklas.
- Ändra mötesrutiner för att alla inom företaget ska veta vad som ska göras på en daglig basis och där man även kan gå igenom vad man gjort och bolla idéer.

Eventuellt implementera roller i projektledningen för att kalla sig ”Scrum”. Inget nödvändigt men skulle kunna vara ett komplement då Scrum redan används till allra största del med undantag för rollerna samt mötesrutinerna.

## 5 Slutsatser

Syftet med arbetet var att utreda Mindbite's val av innehållshanteringssystem och i denna del utvärdera deras egna system och jämföra mot andra och utreda vad som passar dem bäst och varför. Utöver detta skulle jag även utvärdera och gå igenom deras sätt att arbeta i grupp, alltså deras projektledningsmetodik och se hur denna är optimerad för deras verksamhet och komma med eventuella förbättringsförslag.

Arbetet som helhet har fungerat mycket bra, där jag integrerat min praktik och utveckling hos företaget med skrivandet av denna rapport för det självständiga arbetet. Jag har genom den fakta och de frågor som ställts mot företagens anställda tillsammans med praktik kunna dra slutsatser och kommit fram till ett resultat för de frågeställningar och problem som fanns vid ingången av detta projekt.

Arbetet har flutit på och det mesta har gått bra utan några motgångar i själva rapportskrivandet, detta var lätt att undvika på grund av att inget praktiskt kodarbete gjordes specifikt för det självständiga arbetet utan endast teoretiskt varvat med praktik för att få en bättre förståelse för arbetsmetodiken och deras systemutveckling och system.

Arbetet hade kunnat förbättras på ett par punkter, däribland djupare analyser utav CMS:et där en liknande hemsida hade kunnat sättas upp i alla de olika CMS:erna som valdes ut till denna uppgift och utifrån det praktiska moment genomfört analyser av vad som fungerar bäst, varför, hur och vilket som rekommenderas för deras utveckling. Detta hade tagit en stor del tid och hade nog varit orimligt sett till den givna tidsperioden för detta arbete. Förövrigt tycker jag inte att något specifikt i detta arbete hade kunnat göras annorlunda utan endast göra saker och ting på ett djupare plan.

### 5.1 Tillförlitlighet hos utvärderingen utav CMS

Utvärderingen av de utvalda CM-system anser jag som högst tillförlitliga. Detta på grund av insamling av fakta från kvalitativa källor samt intervjuer av kompetenta utvecklare som har erfarenhet utav majoriteten av de valda CMS:en.

Det företaget söker hos ett CMS syns tydligt i ett sådant systems specifikationer och därav kan man dra många slutsatser direkt utifrån just jämförelser utav specifikationerna som gjordes i detta arbete tillsammans med komplementära intervjuer för att få företagens syn på de utvalda innehållshanteringssystemen som inte valdes bort.

För att få en ytterligare del i denna analys, och för att eventuellt förbättra resultatens tillförlitlighet kunde djupare analyser av varje CMS gjorts. Här hade

det dock krävts inläring av de olika CM-systemen samt även titta på dem tillsammans med företagets utvecklare för att även få deras syn på dessa CMS för att få en bra bild över vad som passar dem bäst. Detta hade varit tidskrävande och krävt att utvecklarna på företaget avbrutit deras arbete för att delta i min analys vilket hade varit orimligt både för verksamheten och för arbetets omfattning.

Så med de förutsättningar och den tidsplan som fanns i början av detta arbetet anser jag resultaten som tillförlitliga i den mån möjligt och även användbara för företagets framtida utvecklande vilket jag fick ta del av och de delade även de åsikter jag kom fram till i detta projekt.

## **5.2 Tillförlitlighet hos utvärdering/analys av projektmetodik**

Då jag genom både min praktik samt även här genom intervjuer tagit reda på fakta och upplägg av deras arbete tycker jag att en bra bild över deras sätt att arbeta och bygga upp projekt har lagts upp. Utifrån den insamlade fakten och allt som noterats från min praktik kan man anse som en förstahandskälla på ett sätt och därav tillförlitlig ur den synvinkeln.

Den fakta som har samlats in från källor på internet kring projektledningsmetoderna för att senare dra slutsatser och analyser mot dessa för att se vad som stämmer in bäst och vad som kan förändras i deras projektledning för att göra den optimal har även gått igenom för att undersöka dessa tillförlitlighet. Här har det använts mycket källor från den agila mjukvaruutvecklingens egna principer som finns på flera platser över internet vilka är mycket bra för att få en bild för hur agil projektledning fungerar och därav använts i detta arbete.

För att få ett tillförlitligare resultat och även utöka denna del med ytterligare undersökningar hade man kunnat implementera andra delar i deras projektledning för att se hur resultaten blir och om det fungerar bättre för just dem. Detta är något som hade varit mycket tidskrävande och hade behövts testas under en längre period vilket inte var möjligt. Utöver detta ser jag inte specifika delar som hade kunnat göras ytterligare för att se över deras projektledning utan jag har gjort det möjliga genom att vara en del av företaget och genomföra frågor mot dess anställda.

## **5.3 Förslag till vidareutveckling och forskning**

Detta är en svår del när det gäller vidareutveckling då detta gäller ett specifikt företag. Men vid en analys av liknande delar hos en webbyrå skulle jag rekommendera att under en längre period först delta och göra analyser av projektledningsmetodiken för att sedan göra eventuella optimeringar och se hur det fungerar. Detta kan vara svårt att genomföra men är ett steg djupare än vad som gjorts här.



Gällande CM-systemen hade man kunnat göra en grundligare analys med tester utav CMSerna med inläring och byggande av olika typer av webbplatser för att se vad som är optimalt för vad och vilket som fungerar bäst. För att få en bra bild över vad företaget vill ha hade de behövt delta i denna utvärdering vilket kan vara tidskrävande och därav en svårighet, men som helhet hade även detta varit ett steg längre.

## Källförteckning

- [1] <https://www.codeproject.com/Articles/717523/Top-NET-Based-CMS-Content-Management-System> Hämtad 2017-04-03
- [2] <https://docs.microsoft.com/en-us/aspnet/web-forms/what-is-web-forms> Hämtad 2017-04-03
- [3] [https://en.wikipedia.org/wiki/.NET\\_Framework](https://en.wikipedia.org/wiki/.NET_Framework) Hämtad 2017-04-03
- [4] <https://sv.wikipedia.org/wiki/C-sharp> Hämtad 2017-04-03
- [5] <https://www.ecma-international.org/publications/standards/Ecma-334.htm> Hämtad 2017-04-03
- [6] <https://docs.microsoft.com/en-us/aspnet/core/tutorials/your-first-mac-aspnet> Hämtad 2017-04-03
- [7] <https://www.visualstudio.com/vs/visual-studio-mac/> Hämtad 2017-04-03
- [8] <https://www.codeproject.com/Articles/717523/Top-NET-Based-CMS-Content-Management-System> Hämtad 2017-04-03
- [9] <https://shop.umbraco.com/products-and-support/license/> Hämtad 2017-04-03
- [10] <https://opensource.org/licenses/mit-license.html> Hämtad 2017-04-03
- [11] <https://umbraco.com/pricing/> Hämtad 2017-04-03
- [12] <https://msdn.microsoft.com/en-us/magazine/dn879354.aspx> Hämtad 2017-04-03
- [13] <https://web.archive.org/web/20100916030050/http://umbraco.org/umbraco-4---get-excited/one-cms-any-database> Hämtad 2017-04-03
- [14] <https://umbraco.com/blog/introducing-juno/> Hämtad 2017-04-03
- [15] <https://our.umbraco.org/documentation/reference/templating/masterpages/> Hämtad 2017-04-03
- [16] <https://our.umbraco.org/Documentation/Getting-Started/Setup/Install/install-umbraco-manually> Hämtad 2017-04-03
- [17] <https://our.umbraco.org/documentation/getting-started/setup/install/system-requirements> Hämtad 2017-04-03
- [18] <https://dotnetnuke.codeplex.com/license> Hämtad 2017-04-05
- [19] <https://dotnetnuke.codeplex.com/> Hämtad 2017-04-05
- [20] <http://www.dnnsoftware.com/products> Hämtad 2017-04-05
- [21] <http://www.dnnsoftware.com/compare-dnn> Hämtad 2017-04-05
- [22] <http://www.dnnsoftware.com/platform> Hämtad 2017-04-05
- [23] <http://www.dnnsoftware.com/forums> Hämtad 2017-04-05
- [24] <https://www.microsoft.com/en-us/download/details.aspx?id=5388> Hämtad 2017-04-05
- [25] <http://www.dnnsoftware.com/platform/start/install#2047> Hämtad 2017-04-05

- [26] <http://www.kentico.com/> Hämtad 2017-04-06
- [27] <https://docs.kentico.com/k9/running-kentico-on-microsoft-azure> Hämtad 2017-04-06
- [28] <https://devnet.kentico.com/articles/asp-versus-portal-engine-development-in-kentico-cms> Hämtad 2017-04-06
- [29] <http://www.kentico.com/download-demo/free-cms-for-asp-net> Hämtad 2017-04-06
- [30] <http://www.kentico.com/purchase/price-list/product-license-pricelist> Hämtad 2017-04-06
- [31] <http://www.kentico.com/purchase> Hämtad 2017-04-06
- [32] <http://www.kentico.com/product/all-features> Hämtad 2017-04-06
- [33] <https://docs.kentico.com/k8/developing-websites/choosing-the-right-development-model> Hämtad 2017-04-06
- [34] <https://docs.kentico.com/k9/installation/server-and-hosting-requirements> Hämtad 2017-04-06
- [35] <http://agilemanifesto.org/> Hämtad 2017-04-08
- [36] <https://technology.amis.nl/2007/08/24/agile-software-development-the-principles-2-welcome-change/> Hämtad 2017-04-08
- [37] <https://technology.amis.nl/2007/08/24/agile-software-development-the-principles-2-welcome-change/> Hämtad 2017-04-08
- [38] <https://technology.amis.nl/2007/12/02/agile-software-development-the-principles-principle-3-deliver-working-software-frequently/> Hämtad 2017-04-08
- [39] <https://technology.amis.nl/2007/12/19/agile-software-development-the-principles-principle-4-business-people-and-developers-must-work-together-daily-throughout-the-project/> Hämtad 2017-04-08
- [40] <https://technology.amis.nl/2008/03/02/agile-software-development-the-principles-principle-5-build-projects-around-motivated-individuals/> Hämtad 2017-04-08
- [41] <https://technology.amis.nl/2008/04/23/agile-software-development-the-principles-principle-6-the-most-efficient-and-effective-method-of-conveying-information-to-and-within-a-development-team-is-face-to-face-conversation/> Hämtad 2017-04-08
- [42] <https://technology.amis.nl/2008/07/01/agile-software-development-the-principles-principle-7-working-software-is-the-primary-measure-of-progress/> Hämtad 2017-04-08
- [43] <http://www.agilebuddha.com/agile/agile-principle-simplicity-the-art-of-maximising-the-work-not-done/> Hämtad 2017-04-08
- [44] <https://technology.amis.nl/2009/12/14/agile-software-development-the-principles-principle-11-the-best-architectures-requirements-and-designs-emerge-from-self-organizing-teams/> Hämtad 2017-04-08
- [45] <https://community.dynamics.com/crm/b/crmandunifiedservicedesk/archive/2015/12/01/agile-principle-12-at-regular-intervals-the-team-reflects-on-how-to-become-more-effective> Hämtad 2017-04-08

- [46] [https://www.slideteam.net/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/h/a/ha\\_semicircular\\_arrow\\_for\\_agile\\_development\\_methodology\\_powerpoint\\_template\\_Slide01.jpg](https://www.slideteam.net/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/h/a/ha_semicircular_arrow_for_agile_development_methodology_powerpoint_template_Slide01.jpg)
- [47] <https://www.cprime.com/resources/what-is-agile-what-is-scrum/> Hämtad 2017-04-09
- [48] [http://www.scrum-institute.org/Scrum\\_Roles\\_The\\_Scrum\\_Team.php](http://www.scrum-institute.org/Scrum_Roles_The_Scrum_Team.php) Hämtad 2017-04-09
- [49] <http://www.yodiz.com/blog/product-owner-roles-and-responsibilities-in-agile/> Hämtad 2017-04-09
- [50] <https://www.mountangoatsoftware.com/agile/scrum/roles/scrummaster> Hämtad 2017-04-09
- [51] <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles> Hämtad 2017-04-09
- [52] <https://agilepainrelief.com/notesfromtooluser/2016/10/choosing-the-team-size-in-scrum.html> Hämtad 2017-04-09
- [53] <https://www.scrum.org/resources/what-is-a-scrum-development-team> Hämtad 2017-04-09
- [54] <https://www.neonrain.com/agile-scrum-web-development/> Hämtad 2017-04-09
- [55] [https://s3.amazonaws.com/scrumorg-website-prod/drupal/inline-images/2017-03/SprintBacklog\\_0.png](https://s3.amazonaws.com/scrumorg-website-prod/drupal/inline-images/2017-03/SprintBacklog_0.png)
- [56] <http://wiki.expertiza.ncsu.edu/images/4/43/Scrumteam.gif>
- [57] <https://www.mountangoatsoftware.com/agile/scrum/meetings/daily-scrum> Hämtad 2017-04-09
- [58] <http://www.scrumhub.com/wp-content/uploads/2014/09/standup-meeting.jpg>
- [59] <http://www.scrumhub.com/wp-content/uploads/2014/09/daily-standup.jpg>
- [60] [https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm) Hämtad 2017-04-10
- [61] [http://www.umsl.edu/~hugheyd/is6840/images/Waterfall\\_model.png](http://www.umsl.edu/~hugheyd/is6840/images/Waterfall_model.png)
- [62] <https://atlatz.io/blog/team-roles-in-waterfall-methodology/> Hämtad 2017-04-10
- [63] <http://www.seguetech.com/waterfall-vs-agile-methodology/> Hämtad 2017-04-10
- [64] <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/> Hämtad 2017-04-10
- [65] <http://www.seguetech.com/waterfall-vs-agile-methodology/> Hämtad 2017-04-10
- [66] <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/> Hämtad 2017-04-10
- [67] Figur 2.8, s.20
- [68] <http://www.seguetech.com/waterfall-vs-agile-methodology/> Hämtad 2017-04-12
- [69] <https://1.bp.blogspot.com/-ZyWkTGxYr4M/VwGeo0XiQKI/AAAAAAAAAZ8/zVaimrxkRH0BCzIRZNZDWg4Neg2D4Y3EA/s640/waterfall.png>
- [70] Figur 2.8, s.20

- [71] <http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/> Hämtad 2017-04-12
- [72] <https://choosealicense.com/licenses/mit/> Hämtad 2017-04-12
- [73] <http://umbraco.tv/> Hämtad 2017-04-16
- [74] <http://www.dnnsoftware.com/videos> Hämtad 2017-04-16
- [75] <http://www.dnnstuff.com/articles/articletype/articleview/articleid/118/how-to-make-an-empty-page-template> Hämtad 2017-04-16
- [76] <http://www.dnnsoftware.com/wiki/page-templates> Hämtad 2017-04-16
- [77] <http://umbraco.tv/videos/umbraco-v7/implementor/fundamentals/templating/setting-up-our-first-template/documentation> Hämtad 2017-04-16

## Bilagor

	<b>DNN Platform (formerly DotNetNuke Community Edition) 7.1</b> <input checked="" type="checkbox"/>	<b>Kentico CMS for ASP.NET Free Edition 8.2</b> <input checked="" type="checkbox"/>	<b>Umbraco CMS 4.0</b> <input checked="" type="checkbox"/>
<i>Last Updated</i>	3/24/2015	2/19/2015	2/8/2010
<b>System Requirements</b>	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> <i>Application Server</i>	IIS/.Net	IIS/.Net	IIS/.Net
<input type="checkbox"/> <i>Approximate Cost</i>	\$0	\$0	30EUR + VAT per domain
<input type="checkbox"/> <i>Database</i>	MSSQL	MSSQL	MSSQL
<input type="checkbox"/> <i>License</i>	Open Source	Closed Source	Open Source
<input type="checkbox"/> <i>Operating System</i>	Windows Only	Windows Only	Windows Only
<input type="checkbox"/> <i>Programming Language</i>	C#	C#	C#
<input type="checkbox"/> <i>Root Access</i>	Yes	No	Yes
<input type="checkbox"/> <i>Shell Access</i>	Yes	No	Yes
<input type="checkbox"/> <i>Web Server</i>	IIS	IIS	IIS
<b>Security</b>	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> <i>Audit Trail</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>Captcha</i>	Yes	Yes	Free Add On
<input type="checkbox"/> <i>Content Approval</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>Email Verification</i>	Yes	No	Yes
<input type="checkbox"/> <i>Granular Privileges</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>Kerberos Authentication</i>	Free Add On	Yes	No
<input type="checkbox"/> <i>LDAP Authentication</i>	Free Add On	Yes	Yes
<input type="checkbox"/> <i>Login History</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>NIS Authentication</i>	Free Add On	Yes	No
<input type="checkbox"/> <i>NTLM Authentication</i>	Yes	Yes	No
<input type="checkbox"/> <i>Pluggable Authentication</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>Problem Notification</i>	Yes	No	Yes
<input type="checkbox"/> <i>Sandbox</i>	Yes	No	Yes
<input type="checkbox"/> <i>Session Management</i>	Yes	No	No
<input type="checkbox"/> <i>SMB Authentication</i>	Free Add On	No	No
<input type="checkbox"/> <i>SSL Compatible</i>	Yes	No	Yes
<input type="checkbox"/> <i>SSL Logins</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>SSL Pages</i>	Yes	Yes	Yes
<input type="checkbox"/> <i>Versioning</i>	Yes	No	Yes

Bilaga 1 – CMSMatrix resultat kring systemkrav och säkerhet

<b>Support</b>	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> Certification Program	Yes	No	Yes
<input type="checkbox"/> Code Skeletons	Yes	Yes	Yes
<input type="checkbox"/> Commercial Manuals	Yes	Yes	Yes
<input type="checkbox"/> Commercial Support	Yes	Yes	Yes
<input type="checkbox"/> Commercial Training	Yes	Yes	Yes
<input type="checkbox"/> Developer Community	Yes	Yes	Yes
<input type="checkbox"/> Online Help	Yes	Yes	Yes
<input type="checkbox"/> Pluggable API	Yes	Yes	Yes
<input type="checkbox"/> Professional Hosting	Yes	Yes	Yes
<input type="checkbox"/> Professional Services	Yes	Yes	Yes
<input type="checkbox"/> Public Forum	Yes	Yes	Yes
<input type="checkbox"/> Public Mailing List	Yes	No	Yes
<input type="checkbox"/> Test Framework	Yes	Yes	No
<input type="checkbox"/> Third-Party Developers	Yes	Yes	Yes
<input type="checkbox"/> Users Conference	Yes	No	Yes

Bilaga 2 – CMSMatrix resultat kring support

<b>Ease of Use</b>	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> Drag-N-Drop Content	Yes	Yes	Yes
<input type="checkbox"/> Email To Discussion	No	No	No
<input type="checkbox"/> Friendly URLs	Yes	Yes	Yes
<input type="checkbox"/> Image Resizing	Yes	Yes	Yes
<input type="checkbox"/> Macro Language	Yes	Yes	Yes
<input type="checkbox"/> Mass Upload	Yes	Yes	Free Add On
<input type="checkbox"/> Prototyping	Yes	Yes	Yes
<input type="checkbox"/> Server Page Language	Yes	Yes	Yes
<input type="checkbox"/> Site Setup Wizard	Yes	Yes	Yes
<input type="checkbox"/> Spell Checker	Yes	Yes	Free Add On
<input type="checkbox"/> Style Wizard	Yes	No	No
<input type="checkbox"/> Subscriptions	Yes	No	Yes
<input type="checkbox"/> Template Language	Yes	Yes	Yes
<input type="checkbox"/> UI Levels	Yes	Yes	Yes
<input type="checkbox"/> Undo	Yes	Yes	Yes
<input type="checkbox"/> WYSIWYG Editor	Yes	Yes	Yes
<input type="checkbox"/> Zip Archives	Yes	No	Free Add On

Bilaga 3 – CMSMatrix resultat kring användbarhet och användbar funktionalitet

<b>Management</b>	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> Advertising Management	Yes	No	Free Add On
<input type="checkbox"/> Asset Management	Yes	Yes	Yes
<input type="checkbox"/> Clipboard	Yes	Yes	Yes
<input type="checkbox"/> Content Scheduling	Yes	Yes	Yes
<input type="checkbox"/> Content Staging	Costs Extra	No	Costs Extra
<input type="checkbox"/> Inline Administration	Yes	Yes	Yes
<input type="checkbox"/> Online Administration	Yes	Yes	Yes
<input type="checkbox"/> Package Deployment	Yes	Yes	Yes
<input type="checkbox"/> Sub-sites / Roots	Yes	Yes	Yes
<input type="checkbox"/> Themes / Skins	Yes	Yes	Yes
<input type="checkbox"/> Trash	Yes	Yes	Yes
<input type="checkbox"/> Web Statistics	Yes	No	Yes
<input type="checkbox"/> Web-based Style/Template Management	Yes	Yes	Yes
<input type="checkbox"/> Web-based Translation Management	Yes	No	Yes
<input type="checkbox"/> Workflow Engine	Yes	No	Yes

Bilaga 4 – CMSMatrix resultat kring diverse hantering inom CMSset.

<b>Flexibility</b>	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> CGI-mode Support	Yes	No	No
<input type="checkbox"/> Content Reuse	Yes	Yes	Yes
<input type="checkbox"/> Extensible User Profiles	Yes	Yes	Yes
<input type="checkbox"/> Interface Localization	Yes	Yes	Yes
<input type="checkbox"/> Metadata	Yes	Limited	Yes
<input type="checkbox"/> Multi-lingual Content	Yes	No	Yes
<input type="checkbox"/> Multi-lingual Content Integration	Yes	No	Limited
<input type="checkbox"/> Multi-Site Deployment	Yes	Yes	Yes
<input type="checkbox"/> URL Rewriting	Yes	Yes	Yes

Bilaga 5 – CMSMatrix resultat kring flexibilitet.





Built-in Applications	DNN Platform (formerly DotNetNuke Community Edition) 7.1	Kentico CMS for ASP.NET Free Edition 8.2	Umbraco CMS 4.0
<input type="checkbox"/> Blog	Yes	Yes	Free Add On
<input type="checkbox"/> Chat	Yes	No	No
<input type="checkbox"/> Classifieds	Yes	No	Free Add On
<input type="checkbox"/> Contact Management	Yes	No	No
<input type="checkbox"/> Data Entry	Yes	Yes	Yes
<input type="checkbox"/> Database Reports	Yes	Yes	No
<input type="checkbox"/> Discussion / Forum	Yes	Yes	Free Add On
<input type="checkbox"/> Document Management	Yes	Yes	No
<input type="checkbox"/> Events Calendar	Yes	Yes	Free Add On
<input type="checkbox"/> Events Management	Yes	Yes	Free Add On
<input type="checkbox"/> Expense Reports	Costs Extra	Yes	No
<input type="checkbox"/> FAQ Management	Yes	Yes	Free Add On
<input type="checkbox"/> File Distribution	Yes	No	Free Add On
<input type="checkbox"/> Graphs and Charts	Yes	Yes	No
<input type="checkbox"/> Groupware	Yes	No	Free Add On
<input type="checkbox"/> Guest Book	Yes	No	Free Add On
<input type="checkbox"/> Help Desk / Bug Reporting	Free Add On	No	No
<input type="checkbox"/> HTTP Proxy	Yes	No	No
<input type="checkbox"/> In/Out Board	Free Add On	No	No
<input type="checkbox"/> Job Postings	Yes	Yes	Yes
<input type="checkbox"/> Link Management	Yes	No	Yes
<input type="checkbox"/> Mail Form	Yes	Yes	Yes
<input type="checkbox"/> Matrix	Free Add On	Yes	No
<input type="checkbox"/> My Page / Dashboard	Costs Extra	Yes	Yes
<input type="checkbox"/> Newsletter	Yes	Yes	Yes
<input type="checkbox"/> Photo Gallery	Yes	Yes	Yes
<input type="checkbox"/> Polls	Yes	Yes	Yes
<input type="checkbox"/> Product Management	Yes	Yes	No
<input type="checkbox"/> Project Tracking	Costs Extra	No	No
<input type="checkbox"/> Search Engine	Yes	Yes	Yes
<input type="checkbox"/> Site Map	Yes	Yes	Yes
<input type="checkbox"/> Stock Quotes	Free Add On	No	No
<input type="checkbox"/> Surveys	Yes	Yes	Costs Extra
<input type="checkbox"/> Syndicated Content (RSS)	Yes	No	Yes
<input type="checkbox"/> Tests / Quizzes	Yes	No	No
<input type="checkbox"/> Time Tracking	Costs Extra	No	No
<input type="checkbox"/> User Contributions	Yes	Yes	Free Add On
<input type="checkbox"/> Weather	Free Add On	Free Add On	Free Add On
<input type="checkbox"/> Web Services Front End	Yes	No	No
<input type="checkbox"/> Wiki	Yes	Yes	Free Add On

Bilaga 6 – CMSMatrix resultat kring inbygga applikationer i de olika CMS-systemen.



Bilaga 7 – Mindbite sprintlogg