L’essere umano si è sempre dotato di strumenti che gli permettessero di svolgere diversi compiti (sempre più complessi): l’informatica ha permesso di dotare sistemi autonomi di intelligenza, attraverso una componente software che si è sempre più evoluta.

* L’utilizzo di questi sistemi avviene attraverso la mediazione di interfacce d’uso appositamente progettate e il governo dei sistemi da parte dell’uomo prende sempre più la forma di un dialogo fra due partner intelligenti
  + l’interazione implica sempre più l’esecuzione di ragionamenti complessi da parte di entrambi gli interlocutori.

*Definizioni dell’ISO 9241*

* Per **sistema interattivo** si intende una qualsiasi “*combinazione di componenti hardware e software che ricevono input da un utente umano e gli forniscono un output, allo scopo di supportare l’effettuazione di un compito*”. Questa definizione è molto generale ed esclude solo i sistemi che interagiscono esclusivamente con altri sistemi, senza alcun intervento umano.
* Per **interfaccia d’uso** si intende l’insieme di “*tutti i componenti di un sistema interattivo (hw o sw) che forniscono all’utente informazioni e comandi per permettergli di effettuare specifici compiti attraverso il sistema*”.
  + **Compito** (*task*): qualsiasi insieme di attività richieste per raggiungere un risultato.
* **Dialogo**: “*l’interazione fra un utente e un sistema interattivo, intesa come una sequenza di azioni compiute dall’utente (input) e di risposte del sistema (output), allo scopo di raggiungere un certo obiettivo*”.
  + Può essere realizzato attraverso svariati *dispositivi di interazione* (dispositivi di input e di output).

### Dimensioni della complessità

Un sistema può essere considerato complesso per aspetti diversi:

* **complessità interna** (o **strutturale**): molti componenti che interagiscono fra loro in modo complicato
* **complessità esterna** (o **funzionale**): numerose attività supportate

Le moderne applicazioni software possono raggiungere entrambi gradi di complessità elevatissimi: per misurarli sono state definite varie metriche (rispettivamente, righe di codice sorgente e *punti funzione* per misurare le funzionalità visibili all’utente, indipendenti dalla particolare tecnologia con cui vengono realizzate).

La **complessità d’uso** di un sistema corrisponde alla facilità con cui si è in grado di utilizzarlo (bassa se il sistema è facile da usare): è un concetto indipendente dai primi due e deve tenere in considerazione la *diversità degli utenti*.

Ogni utente si rapporta con il sistema in modo specifico e chiede ad esso di supportarlo nell’esecuzione di compiti specifici. I problemi legati alla complessità d’uso non sono intrinseci allo strumento ma derivano dall’interazione fra strumento e utente; il problema di disporre di interfacce d’uso adeguate diviene critico, dal momento in cui queste interfacce devono risolvere problemi complessi di utenti tra loro molto diversi.

### Ruolo dell’interfaccia utente

L’accelerazione del progresso tecnologico influenza ed è influenzata dai bisogni degli utenti e dalla forte competitività e concorrenza sul mercato. Prodotti dell’informatica e delle telecomunicazioni costituiscono un vero e proprio ecosistema in perenne evoluzione (cambiamenti di un prodotto producono la necessità di cambiamenti adattativi nei prodotti a esso correlati).

Una delle conseguenze dello sviluppo tecnologico è la forte tendenza all’*iperfunzionalismo*: prodotti sul mercato tendono a fornire prestazioni in eccesso rispetto alle esigenze degli utenti. Questo è considerato un:

* *vantaggio* per l’utente, che può trovare sul mercato soluzioni adeguate per le esigenze più varie
* *problema* per chi ha esigenze “normali”, che deve districarsi tra funzioni a lui non necessarie. E’ fondamentale che la tecnologia sia ugualmente accessibile a tutti coloro che ne possano beneficiare.

Un approccio complementare all’alfabetizzazione e istruzione digitale degli utenti prevede la *modifica della tecnologia dall’interno, promuovendo fra chi la progetta e la produce una cultura della semplicità*, che consideri la facilità d’uso come un prerequisito indispensabile.

Spesso in ingegneria del software, infatti, l’interfaccia utente è vista come un “abbellimento dell’ultimo minuto”; dal punto di vista dell’utente finale, essa assume invece il ruolo più importante.

La tecnologia deve adeguarsi alla popolazione, non il vice-versa: è responsabilità dei progettisti acquisire le conoscenze e gli strumenti per la progettazione di sistemi utilizzabili senza problemi dai principali utenti.

* L’interfaccia utente è l'elemento di comunicazione tra l’utente e il sistema, il mezzo con cui l’utente interagisce al fine di raggiungere i suoi obiettivi. Costituisce la parte più importante del sistema dal punto di vista dell’utente.
* L’interfaccia d’uso ha il compito di “filtrare” la complessità”, presentando all’utente un’immagine semplificata del prodotto
* Essa deve mettere a disposizione dell’utente funzioni di più alto livello, riducendo così la *complessità funzionale*: ciò viene realizzato integrando numerose funzionalità semplici in altre più potenti, semplificando il dialogo tra utente e sistema.

*Nota: questo ruolo di semplificazione produce come conseguenza una significativa crescita quantitativa del software che la gestisce (45-50%).*

La sfida del progettista consiste nel conciliare complessità e semplicità d’uso, per tutti gli utenti: può essere vinta solo modificando completamente l’approccio tradizionale alla progettazione dei sistemi, introducendo un nuovo paradigma nell’ingegneria (progettazione centrata sull’essere umano, che consideri le esigenze dell’utente prima di ogni altra cosa).

### Human-Computer Interaction

Questo ruolo dell’interfaccia come strumento di semplificazione assume importanza rilevante con la diffusione dei prodotti tecnologici sui mercati di massa:

* Nel 1981 IBM lancia sul mercato il suo PC: i nuovi prodotti software si rivolgono a una massa di utilizzatori senza competenze di informatica

Sorge una disciplina nuova, **Human-Computer Interaction** (HCI): “*si occupa della progettazione, valutazione e realizzazione di sistemi interattivi basati su computer destinati all’uso umano e dello studio dei principali fenomeni che li circondano.*” (Da ACM SIGCHI

Curricula for Human Computer Interaction).

* Nel 1982 nasce il SIGCHI, *Special Interest Group for Computer-Human Interaction* dell’ACM, l’associazione dei professionisti americani dell’informatica.
* Nel 1983 nasce la *Computer-Human Interaction Conference* dell’ACM, organizzata annualmente dal SIGCHI
* Nel 1984 lnteract Conference dell’IFIP; nel 1985 la HCI Conference della British Computer Society

L’HCI è un’area interdisciplinare:

* *computer science* → progettazione delle applicazioni e ingegnerizzazione delle interfacce umane
* *psicologia* → l’applicazione delle teorie dei processi cognitivi e l’analisi empirica dei comportamenti degli utenti. Interessano leggi sperimentali che indicano come il sistema visivo degli esseri umani interpreta quello che vede e con che facilità interpreta i messaggi. In questo modo si possono fornire delle linee guida per i progettisti.
* *sociologia* e *antropologia* → interazioni fra tecnologia, lavoro e organizzazione
* *industrial design* → prodotti interattivi

Queste discipline hanno background, caratteristiche e problematiche molto diverse: progettare una buona interfaccia utente richiede che le diverse prospettive (in particolare quelle riguardanti l’informatica e la psicologia) vengano messe insieme.

L’HCI ha le sue origini in due aree disciplinari molto diverse:

* l’**ergonomia** (ergon: lavoro, nomos: legge):
  + Studia l’attività umana in relazione alle condizioni ambientali, strumentali e organizzative in cui si svolge, col fine di adattarle alle esigenze dell’uomo.
    - Affronta soprattutto la disposizione ottimale dell’ambiente e delle apparecchiature di lavoro in funzione dei compiti da svolgere
  + Studia le norme che tutelano la vita del lavoratore e accrescono l’efficienza e l’affidabilità dei sistemi uomo-macchina.
  + L’obiettivo attuale è contribuire alla progettazione di oggetti e ambienti di lavoro che rispettino i limiti dell’uomo e ne potenzino le capacità operative.
    - E’ concentrata sullo studio dei processi cognitivi e di elaborazione delle informazioni sottostanti ai processi del lavoro umano
* la **scienza dei computer**: in particolare si sperimentano nuove interazioni con la macchina, nella convinzione che gli elaboratori possano diventare delle protesi cognitive, utilizzabili per eseguire compiti complessi e *“aumentare l’intelletto umano”* (empower).

**Temi dell’HCI**

* Metodologie e processi per la progettazione delle interfacce fra uomo e computer
* Metodi e strumenti per la loro realizzazione
* Tecniche per la valutazione e il confronto di interfacce
* Progettazione di nuove tecniche d’interazione
* Sviluppo di modelli descrittivi e predittivi dell’interazione

Si può puntualizzare che in generale è cambiato l’oggetto di studio dalla disciplina: dall’interazione con i computer in senso stretto, oggi si rivolge a quello dell’interazione con oggetti interattivi di ogni tipo.

*Human centered artificial intelligence*: la progettazione di sistemi AI che aumentino le capacità e il potere agli esseri umani vede l’HCI come fondamentale, studiando attraverso questa disciplina gli obiettivi delle persone e il loro comportamento nella società o nel contesto di lavoro.

### Usabilità

E’ utile analizzare il modo con cui si interagisce con un sistema interattivo per individuare dove nasce la complessità d’uso. Il modello più semplice dell’interazione fra un sistema e il suo utilizzatore è rappresentato dal **ciclo di feedback** (*feedback loop*):

* L’utente, per raggiungere il proprio scopo, fornisce un input al sistema e riceve da questo una risposta (*feedback*), che viene interpretata e confrontata con lo scopo iniziale.
* Il risultato di questo confronto porta alla successiva azione dell’utente, innescando così un nuovo ciclo.

*Nota: per una maggiore usabilità, il feedback dev’essere facilmente interpretabile dall’utente, cioè deve indicare chiaramente quali modifiche siano state prodotte dall’input verso il sistema. Il feedback dev’essere ben comprensibile e specifico (importante la tempestività: se la distanza temporale con la sua azione è significativa, i due eventi potrebbero essere interpretati come tra loro indipendenti. E’ possibile aggiungere feedback intermedi oppure mostrare, quando possibile, una stima quantitativa del tempo mancante).*

Il termine **affordance** denota la proprietà di un oggetto di influenzare, attraverso la sua apparenza visiva, il modo in cui viene usato.

Ci interessa una definizione operativa dell’**usabilità** di un sistema, che permetta di quantificarla dandone (per quanto possibile) una misura oggettiva.

**Usabilità nell’ISO 9241:** si occupa dell’interazione tra le persone e il calcolatore. La specifica 9241-11 (parte 11 del documento) si occupa di definire l’usabilità (definizione più generale):

*La misura in cui un sistema, prodotto o servizio può essere usato da specifici utenti per raggiungere specifici obiettivi con efficacia, efficienza e soddisfazione in uno specifico contesto d’uso.*

*Nota: in una definizione precedente, le parole in rosso non c’erano (si parlava poco di software services).*

Questa definizione è multidimensionale, che scompone l’usabilità su tre assi sostanzialmente indipendenti:

* **Efficacia**: accuratezza, completezza e assenza di conseguenze negative (danni economici, alla salute, all’ambiente) con cui gli utenti raggiungono specifici obiettivi. Considera quindi il “livello di precisione” con cui l’utente riesce a raggiungere i suoi scopi.
* **Efficienza**: le quantità di risorse spese per raggiungere specifici obiettivi (es. tempo, numero di tasti da premere, numero di operazioni).
* **Soddisfazione**: il comfort e il piacere nell’uso del sistema (tiene conto anche di aspetti importanti per UX, come l’attitudine positiva).

Queste grandezze devono essere definite caso per caso, in base alla natura dello specifico sistema: l’usabilità non è una proprietà assoluta dei sistemi, ma dipende dagli utenti, dagli obiettivi e dal contesto d’uso (aspetto puntualizzato dallo stesso standard). Anche l’importanza relativa delle tre grandezze individuate dev’essere valutata caso per caso, in funzione degli obiettivi del sistema.

### Apprendibilità e memorabilità

Serve approfondire la nozione di usabilità prendendo in considerazione l’evoluzione che può subire un utente nel tempo nella sua relazione con il sistema: all’inizio non lo conosce affatto (*utente novizio*), poi inizia a usarlo (*principiante*), fino a diventare *competente* ed *esperto.*

* Prodotti molto simili per quanto riguarda le funzioni offerte possono avere *profili di apprendimento* molto diversi.
* Un sistema facile da imparare si dice dotato di elevata **apprendibilità** (*learnability*).

Nella progettazione di un sistema è possibile considerare come principali destinatari del prodotto gli utenti occasionali (che non sono disposti a investire tempo in attività di apprendimento) o gli utenti continuativi (il cui obiettivo è imparare a usare il prodotto con la massima efficacia ed efficienza). A seconda di ciò, i risultati della progettazione saranno molto differenti; una terza possibilità è quella di fornire entrambi i profili di apprendimento, tramite

* funzioni di rapido apprendimento per gli utenti occasionali
* funzioni di più lento apprendimento, che permettono di ottenere gli stessi risultati con maggiore efficienza o efficacia.

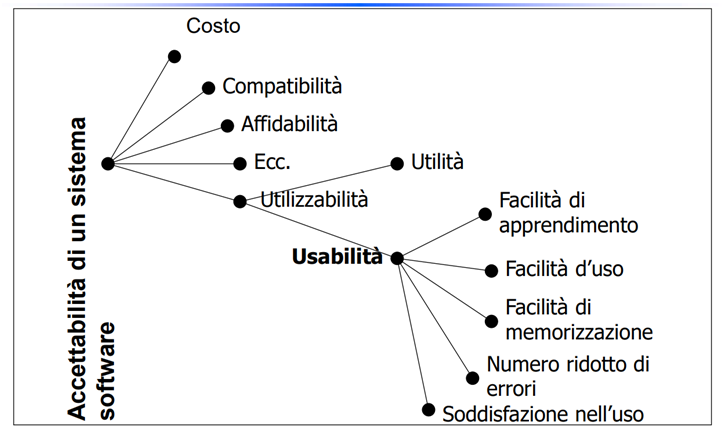
Nel caso di utenti occasionali, è utile che le modalità d’uso del prodotto siano facili da ricordare (**memorability**)

### Usabilità: definizione di Nielsen

Jakob Nielsen (1992) definisce l’accettabilità da parte dell’utente finale di un sistema software e evidenzia le sue componenti: accanto alle classiche qualità su cui si focalizza l’ingegneria del software (costo software, compatibilità con altri sistemi, affidabilità, …), egli afferma che l'accettabilità dipende da quanto un sistema è *utilizzabile* dall’utente.

L’utilizzabilità viene a sua volta divisa in due dimensioni:

* **utilità**: valuta quanto il prodotto è utile per gli scopi dell’utente
* **usabilità:** definita come un concetto multidimensionale, caratterizzato da 5 dimensioni. Il prodotto software:



* dev’essere facile da apprendere (*facilità di apprendimento*)
* dev’essere facile da usare (*facilità d’uso*)
* le sue funzionalità devono essere chiare e devono permettere di ricordare facilmente le modalità di interazione (anche dopo molto tempo di disuso)
* non deve indurre gli utenti a fare errori; deve anzi ridurre gli errori che l’utente può commettere (es. limitare la possibilità dell’utente di usare la tastiera, attività fortemente prona agli errori) e permettere di recuperare facilmente qualora avvengano.
* deve anche dare soddisfazione nell’uso all’utente: dev’essere piacevole da usare e dare appagamento.

*Differenze tra le definizioni:* La definizione data nell’ISO 9241 è in grado di tenere in considerazione anche gli effetti di apprendibilità e memorabilità. In generale:

* soddisfazione d’uso in comune alle due definizioni
* l’efficienza dipende dalla facilità d’uso e dai pochi errori
* mentre la facilità di apprendimento e di memorizzazione intervengono sull’efficacia.

Le definizioni enfatizzano dimensioni diverse ma in realtà la sostanza è abbastanza simile.

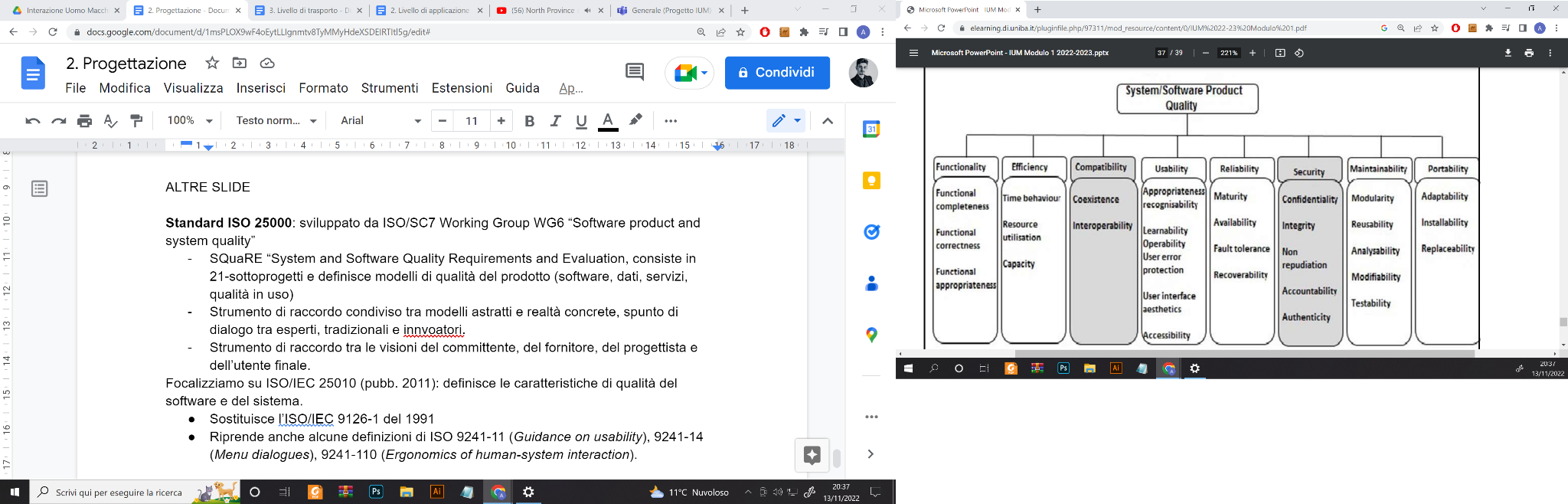
### Usabilità: definizione standard 25010

**Standard ISO 25000**: sviluppato da ISO/SC7 Working Group WG6 “Software product and system quality”

* SQuaRE “System and Software Quality Requirements and Evaluation, consiste in 21-sottoprogetti e definisce modelli di qualità del prodotto (software, dati, servizi, qualità in uso)
* Strumento di raccordo condiviso tra modelli astratti e realtà concrete, spunto di dialogo tra esperti, tradizionali e innovatori.
* Strumento di raccordo tra le visioni del committente, del fornitore, del progettista e dell’utente finale.

Focalizziamo su ISO/IEC 25010 (pubb. 2011): definisce le caratteristiche di qualità del software e del sistema.

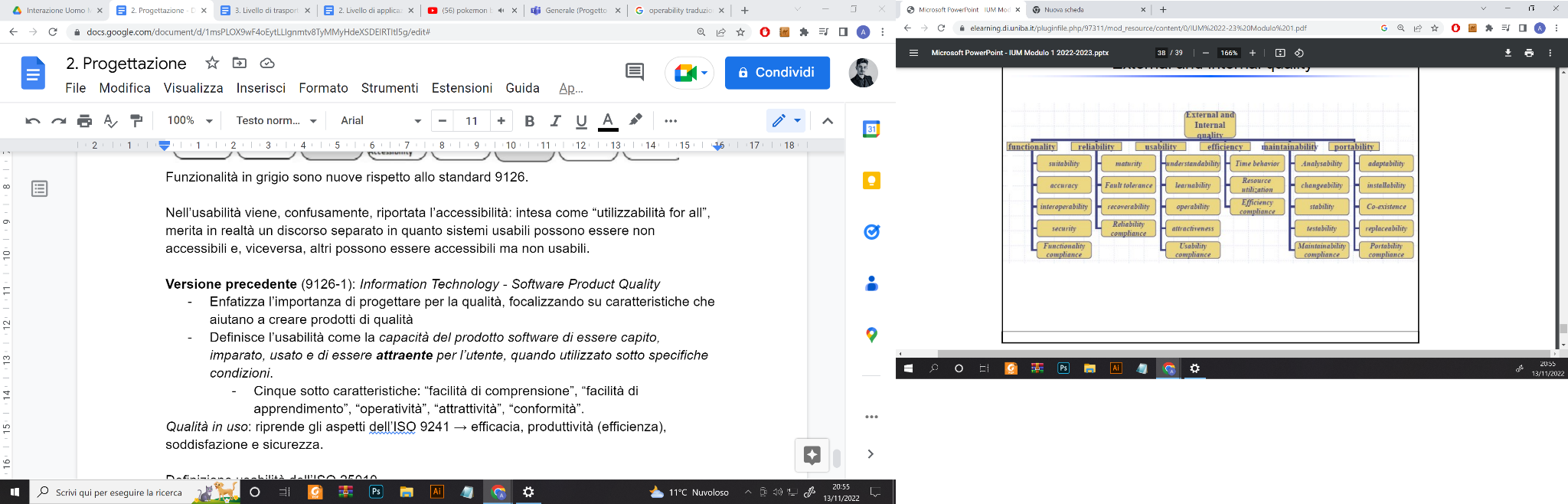
* Sostituisce l’ISO/IEC 9126-1 del 1991
* Riprende anche alcune definizioni di ISO 9241-11 (*Guidance on usability*), 9241-14 (*Menu dialogues*), 9241-110 (*Ergonomics of human-system interaction*).

Funzionalità in grigio sono nuove rispetto allo standard 9126.

Nell’usabilità viene, confusamente, riportata l’accessibilità: intesa come “utilizzabilità for all”, merita in realtà un discorso separato in quanto sistemi usabili possono essere non accessibili e, viceversa, altri possono essere accessibili ma non usabili.

**Versione precedente** (9126-1): *Information Technology - Software Product Quality*

* Enfatizza l’importanza di progettare per la qualità, focalizzando su caratteristiche che aiutano a creare prodotti di qualità
* Definisce l’usabilità come la *capacità del prodotto software di essere capito, imparato, usato e di essere* ***attraente*** *per l’utente, quando utilizzato sotto specifiche condizioni*.
  + Cinque sotto caratteristiche: “facilità di comprensione”, “facilità di apprendimento”, “operatività”, “attrattività”, “conformità per l’usabilità”.



*Qualità in uso*: riprende gli aspetti dell’ISO 9241 → efficacia, produttività (efficienza), soddisfazione e sicurezza.

### User Experience

L’ingegneria dell’usabilità si occupa dello sviluppo di tecniche per valutare l’usabilità: *Ben Shneiderman* ha proposto di “misurare” i progressi della HCI sulla base del numero di persone che possono fare una determinata cosa e del tempo richiesto per farla.

es. Fare un ritratto

* 200 anni fa disegno: poche persone 1 ora
* oggi col cellulare: milioni di persone, 1 secondo

*Obiettivo:* progettare sistemi usabili (= interfaccia usabile), in grado di far vivere negli utilizzatori una buona user experience (include l’usabilità, infatti un sistema difficile da usare difficilmente darà una buona UX).

Negli ultimi anni, il concetto di usabilità si è evoluto: l’HCI si occupa di user experience, focalizzando aspetti estetici ed emotivi nella progettazione dei sistemi.

**Definizione nell’ISO 9241-210**

**UX:** *le percezioni e le relazioni di un utente che derivano dall’uso o dall’aspettativa d’uso di un prodotto, sistema o servizio*

La UX si focalizza su

* emozioni
* preferenze
* reazioni psicologiche e fisiche
* i comportamenti che si verificano prima, durante o dopo l’utilizzo del sistema

realizzando una componente fondamentale della *soddisfazione d’uso*, che caratterizza l’usabilità. Questa è fattorizzata in attributi elementari collegati alle emozioni, alla sorpresa, alla gioia e al divertimento, ecc.

* Anche l’esperienza d’uso è influenzata dal sistema, dall’utente e dal contesto d’uso (quest’ultimo influenza molto L’I/O di un sistema interattivo, ad esempio laddove ci sono rumori serve stare attenti all’input/output sonoro).
* Un progetto pensato (fin dall’inizio) per dare una user experience di alta qualità consiste nel pensare anche a quanto il prodotto possa creare piacere negli utilizzatori.
* Con la user experience, l’enfasi sulle emozioni è maggiore. Aspetti desiderabili: “divertente”, “coinvolgente”, “stimolante”. Anche qui dipende dal tipo di sistema (es. gioco genera stress, eccitazione) anche se, in generale, non dev’ essere noioso o frustrante.

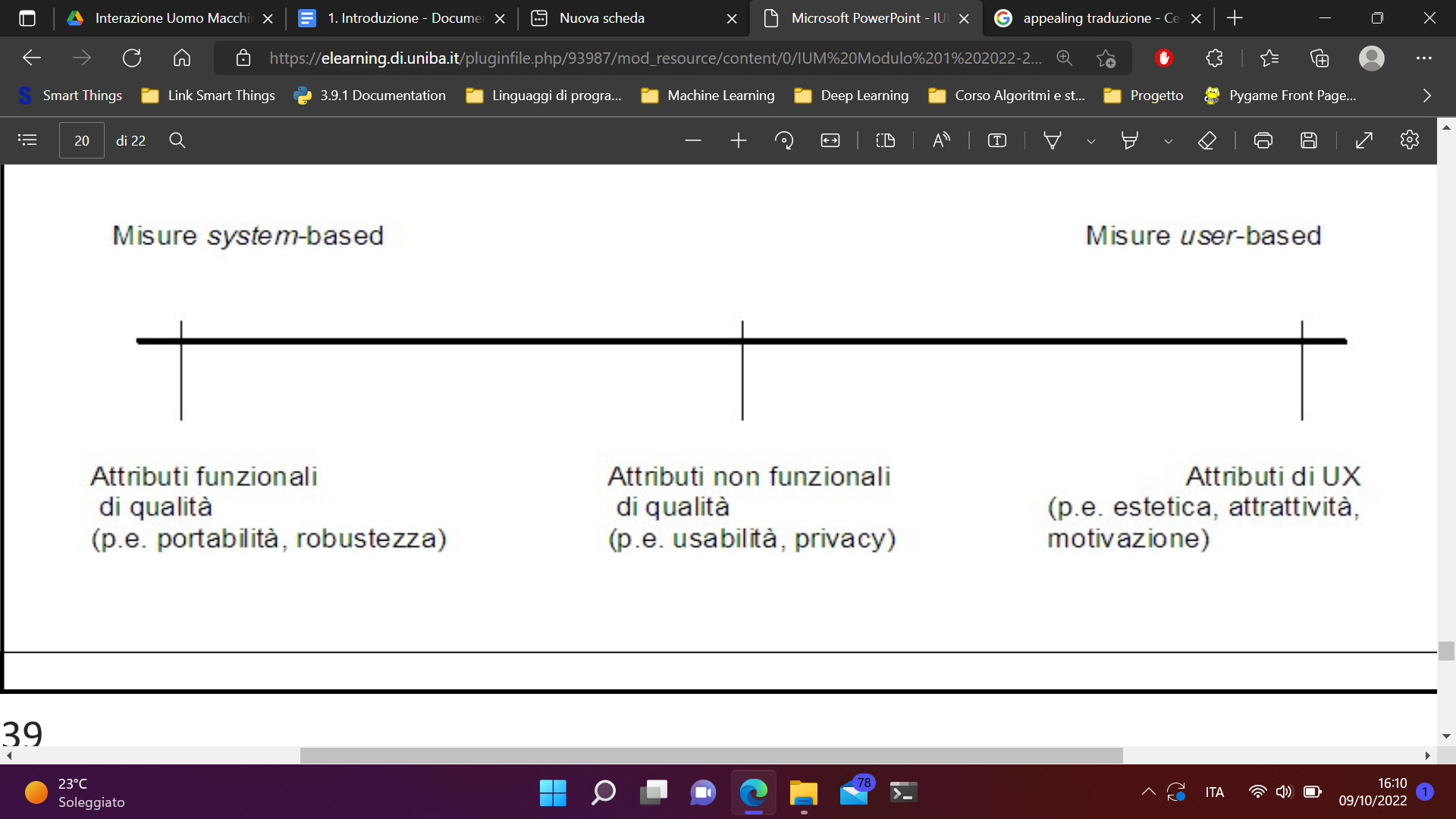
*Donald Norman: “fino ad ora obiettivo primario della progettazione del prodotto o servizio è fornire funzioni utili” … “anche attenzione su creare prodotti piacevoli, emozioni componenti principali del progetto, dobbiamo anche incorporare azioni, in pratica aprirsi” … “impariamo da esperienze complete, non da astrazioni. L’astrazione viene dopo”.*

***Da usabilità a UX***

Progettare per UX include molto di più che progettare per efficienza e altri attributi tradizionali.

* Efficienza: veloce, facile, funzionale, senza errori
* UX: coinvolge sensazioni e si focalizza sul lato emotivo, stimolazione bello e anche su sensazioni tattili o acustiche.

L’esperienza di un utente con un prodotto è influenzata da attributi di qualità del prodotto (funzionali o non funzionali) e da attributi specifici della user experience (piacere, desiderabilità).



*Evoluzione di prodotti innovativi nelle industrie*:

1. inizialmente si progetta per l’utilità
2. man mano che si evolve l’industria, si pone l’attenzione sull’usabilità.
3. Quando la tecnologia è matura, l’attenzione và verso la UX

Un prodotto che genera una UX positiva dovrebbe essere utile, usabile e desiderabile. Per creare prodotti desiderabili, la UX pone enfasi su piacere, estetica e divertimento.

* Alcuni studi mostrano la correlazione tra estetica e percezione della qualità dell’interfaccia: interfacce esteticamente accattivanti sono considerate più utili anche se lo sono meno di altre con funzionalità simili ma meno accattivanti
* Uno studio riporta che l’appropriatezza di dettagli grafici contribuisce a una buona UX.

Per migliorare l’interfaccia utente:

* convincere gli ingegneri del software dell’importanza della sua qualità, in modo tale che le conoscenze in HCI li guidino nelle loro decisioni (progettare una buona interfaccia deve diventare un problema stimolante come il progetto di codice ben strutturato)
* integrare metodi e tecniche di progetto per le interfacce nelle metodologie standard di sviluppo del software nonché adottare procedure di testing e di controllo di qualità per interfacce analoghe a quelle già accettate per valutare altri aspetti del sistema.

Non esistono ricette precise, serve iniziare utilizzando linee guida: non c’è alcun sostituto al testing.

### Valutazione dei sistemi software

Talvolta valutazione e testing sono usate come sinonimo. In generale, la valutazione ha un significato più ampio e quando se ne parla in HCI include un test che coinvolge gli utenti (provano le funzionalità del sistema).

Serve tener conto che il progetto va sottoposto a valutazioni, per verificare che il sistema risponda in modo adeguato alle necessità degli utenti (requisiti come efficacia, facilità di apprendimento, facilità d’uso).

Gli obiettivi della valutazione (*in ingegneria del software*) sono le funzionalità del sistema, la sua usabilità e la user experience; il processo va applicato alla fine dello sviluppo del sistema per verificare la corrispondenza con i requisiti originali del progetto.

* Effettuata esclusivamente da sviluppatori o esperti del dominio
* Valutazione di aspetti limitati del sistema, scarsa attenzione sulla valutazione dell’usabilità.

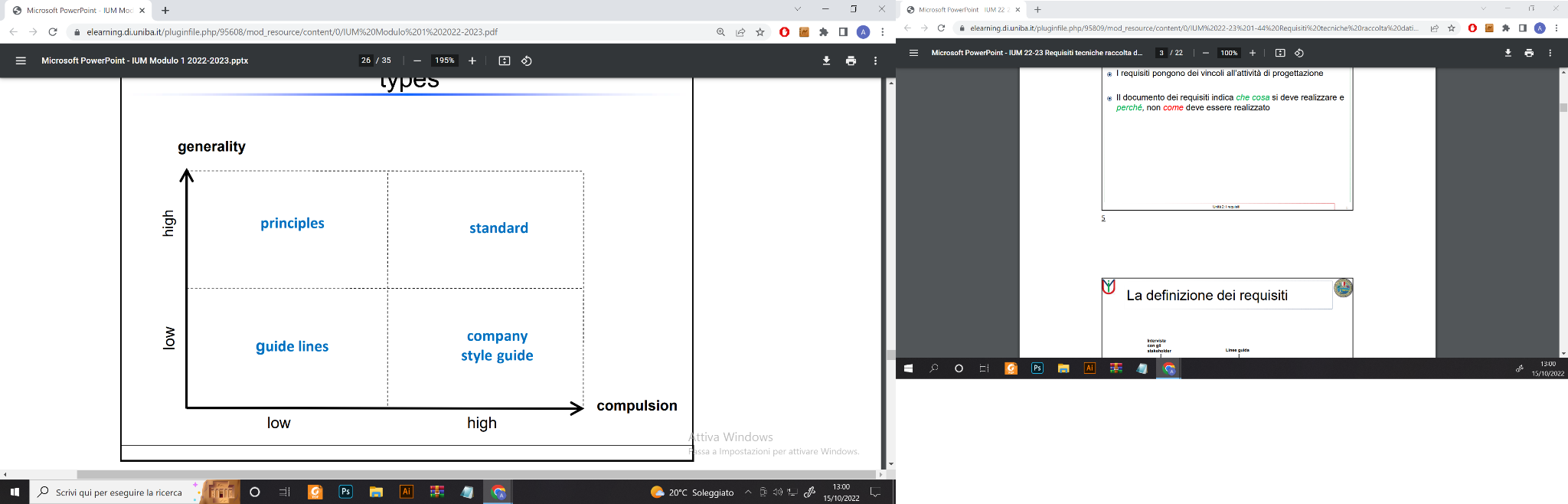
La valutazione è uno strumento essenziale per migliorare il progetto e le performance di un sistema e deve invece focalizzarsi sull’user experience al pari che sulle funzionalità del sistema.

### Indicazioni per la progettazione

Per aiutare il progettista di sistemi interattivi usabili, è utile fornirgli delle indicazioni che si sono dimostrate valide in progetti simili al suo: ce ne sono di vari tipi, distinte su due dimensioni

* **generalità**: alcune indicazioni sono applicabili in ogni situazione, altre in casi specifici
* **obbligatorietà:** alcune sono semplici suggerimenti, altre sono vincolanti e devono essere seguite obbligatoriamente.

Le indicazioni possono essere classificate in quattro grandi categorie.



**Linee guida a livello di compagnia**: sono imposte dal committente e di solito sono regole piuttosto dettagliate, che specificano le modalità di interazione con un certo sistema. In genere hanno l’obiettivo di rendere coerenti i sistemi di una stessa organizzazione.

* Esempi tipici: regole che definiscono l’apparenza grafica dell’UI (font, forme, colori, schemi ricorrenti, …).

**Standard**: norme di tipo generale emesse da organismi internazionali (ISO - *International Standard Organization*) o enti indipendenti (W3C - *World Wide Web Consortium*), le cui raccomandazioni costituiscono degli standard *di fatto*.

Sono regole da applicare nei progetti di determinate classi di sistemi e sono vincolanti per i progetti che dichiarano di esserne conformi.

La conformità a uno standard deve essere valutabile in modo preciso, per cui gli standard devono essere formulati con estrema cura, per evitare ambiguità.

**Linee guida**: sono raccomandazioni per specifiche classi di sistemi, espresse in modo più o meno generale e spesso corredate di esempi e motivazioni.

Particolarmente importanti sono le linee guida per l’interfaccia utente di applicazioni relative a una specifica piattaforma, il cui scopo principale è garantire un’elevata uniformità della user experience di tutte le applicazioni sviluppate per quella piattaforma.

**Principi**: indicazioni generali per la progettazione di interfacce utente usabili, basate su evidenze scientifiche o sul generale consenso, che derivano dalla conoscenza di aspetti psicologici, fisiologici e sociali degli utenti. Sono indipendenti dalla tecnologia e sono espressi spesso in forma molto generale (principi del dialogo con i sistemi interattivi descritti nell’ISO 9241-110).

### Euristiche di Nielsen

Esempi di principi per l’usabilità sono le **euristiche di Nielsen** (versione aggiornata, non sono i principi del libro del 1993, ma ampliati negli anni). Un progettista deve conoscere questi principi ed essere in grado di applicarli sia nella progettazione che nella valutazione.

1. **Visibilità dello stato del sistema**: il sistema dovrebbe mantenere l’utente informato su quello che sta succedendo, attraverso feedback appropriati e in tempi ragionevoli.

* Es: invece di lasciare la schermata ferma (l’utente si chiede che sta succedendo), formare una barra di progresso (il *feedback* è un’indicazione di cosa il sistema sta facendo) per dare all’utente la sensazione che procede tutto correttamente.

1. **Corrispondenza tra sistema e mondo reale**: il sistema dovrebbe parlare il linguaggio degli utenti (utilizzare parole, frasi e concetti familiari all’utente piuttosto che system-oriented).

* Seguire convenzioni del mondo reale e fornire informazioni in un ordine naturale o logico.
* Es: i messaggi di errore sono esempi di indicazioni poco usabili: si deve cercare di utilizzare un linguaggio familiare all’utente e dare indicazioni su come recuperare dall’errore, descrivendo i passi da compiere.
* Es: icone che riproducono situazioni familiari nella vita dell’utente.
* Es: i menù devono essere progettati coerentemente con l’ordine logico delle azioni (in alto a sinistra ci sono le informazioni per eseguire operazioni più frequenti).

1. **Controllo e libertà dell’utente**: progettare interfacce utente in cui l’utente deve avere la sensazione di avere il controllo. Egli può fare degli errori, per cui serve fornire modi per risolvere velocemente il problema (uscite di emergenza, *undo/redo*).
2. **Coerenza e standard**: l’utente non dovrebbe chiedersi se parole differenti, situazioni o azioni indicano la stessa cosa. Coerenza e standard implicano una maggiore chiarezza nonché facilità di apprendimento.

* Le interfacce iniziali di alcune applicazioni sono simili: la loro progettazione è coerente e questo aiuta l’utente facilitando l’apprendimento.

1. **Prevenzione degli errori**: progettare per minimizzare la possibilità degli utenti di compiere errori, per cui serve evitare condizioni error-prone.

* Es: preferire menu a tendina rispetto alla scrittura
* Es: chiedere all’utente conferma delle proprie azioni

*Nota: nei principi visti non si fa esplicitamente riferimento a interfacce visuali (nel 6 si)*

1. **Riconoscere invece di ricordare**: minimizzare il ricorso alla memoria dell’utente rendendo visibili oggetti, azioni e opzioni. L’interfaccia grafica ha successo perché riconoscere è più facile che ricordare Cercare inoltre di rendere sempre visibili le opzioni più ricorrenti e importanti, in modo che siano facilmente eseguibili

* Questo principio è un esempio di conoscenza dei fattori umani, caratterizza multidisciplinarietà dell’HCI.

1. **Flessibilità ed efficienza d’uso**: permettere agli utenti più esperti di utilizzare acceleratori (per diversi comandi), in modo da incrementare l’efficienza. Si noti che gli utenti novizi possono trascurare questi acceleratori e la loro presenza permette a ogni utente di personalizzare le azioni più frequenti.
2. **Attenzione all’estetica e design minimalista**: l’estetica può confondere, in quanto le animazioni attirano l’attenzione dell’essere umano (dalla psicologia) portando distrazioni. La progettazione della schermata non deve sovraccaricare l’utente né contenere informazioni irrilevanti.

* Funzionalità usate raramente faranno parte di sottomenu e saranno nascoste dal menu principale, per essere mostrate solo all’occorrenza.
* La presenza di più unità di informazione diminuisce il valore delle unità rilevanti, penalizzandole.
* *Nota: questo principio non c’era nella prima stesura, in quanto negli anni ‘90 non c’era particolare attenzione all’estetica. In seguito ci si è resi conto di quest’aspetto confrontando sistemi concorrenti.*

1. **Aiutare utenti a riconoscere, diagnosticare e recuperare dagli errori**: gli errori devono essere espressi in linguaggio semplice (senza codici), indicare precisamente il problema e suggerire in modo costruttivo la soluzione.
2. **Aiuto e documentazione**: sebbene è meglio che il sistema sia usato senza documentazione, può essere necessario fornire aiuto e documentazione. Ogni informazione dovrebbe essere facilmente raggiungibile e dovrebbe elencare i passi concreti da fare.

* Esempio: è stato dimostrato che molti incidenti di aerei sono dovuti alla cattiva interfaccia a disposizione del pilota.

Recenti problemi su 737 (riguardanti le AI, che hanno un certo grado di precisione) sono dovuti a un mancato controllo dei piloti in quanto non erano stati addestrati per interfacciarsi col sistema.

Anche un altro pioniere dell’HCI (Shneiderman) individua principi per la progettazione di interfacce (8, simili)

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialog to yield closure*
5. *Offer simple error handling*
6. *Permit easy reversal of action*
7. *Support internal locus of control*
8. *Reduce short-term memory load:* rendere visibili le operazioni più frequenti per minimizzare lo sforzo

Anche Donald Norman, che ha introdotto la modalità di progettazione user centered, individua principi di progettazione quali *visibilità* (stato del sistema), *feedback*, *affordance*, *mapping* (dare indicazioni su quello che si può fare), *vincoli* (guidare l’utente rendendo non disponibili operazioni in momenti specifici, riducendo errori) e *coerenza.*

**Principi ISO 9241-110** (*principi per il dialogo*)

Questo documento descrive sette caratteristiche che ogni dialogo fra un utente e un sistema interattivo dovrebbe possedere. Si riportano le euristiche di Nielsen corrispondenti a ogni principio.

* *Adeguatezza al compito*: il dialogo dovrebbe supportare l’utente nel completamento del compito (le funzionalità del sistema e il dialogo sono basati sulle caratteristiche del compito).
  + [riconoscere piuttosto che ricordare, flessibilità ed efficienza d’uso]
* *Autodescrizione*: il dialogo deve rendere evidente agli utenti il punto in cui si trovano e quali azioni possono compiere per andare avanti.
  + [visibilità dello stato del sistema]
* *Controllabilità:* l’utente dovrebbe essere in grado di iniziare e gestire la direzione e i tempi d’interazione fino al raggiungimento dei suoi obiettivi.
  + [libertà e controllo da parte degli utenti]
* *Conformità alle aspettative dell’utente*: il dialogo deve corrispondere alle necessità dell’utente, prevedibili in base al contesto.
  + [corrispondenza mondo reale e sistema; coerenza e standard]
* *Tolleranza verso gli errori*: nonostante le inesattezze negli input, i risultati desiderati devono poter essere ottenuti senza (o con minime) azioni correttive. Questa caratteristica si ottiene attraverso controllo, correzione e gestione degli errori.
  + [prevenzione degli errori, aiutare gli utenti a riconoscere errori ..]
* *Adeguatezza all’apprendimento:* il dialogo deve supportare e guidare l’utente nell’apprendimento del sistema.
  + [aiuto e documentazione]
* *Adeguatezza all’individualizzazione*: l’utente deve poter modificare l’interazione e la presentazione delle informazioni per adattarle alle proprie necessità individuali.

### User-Centered Design

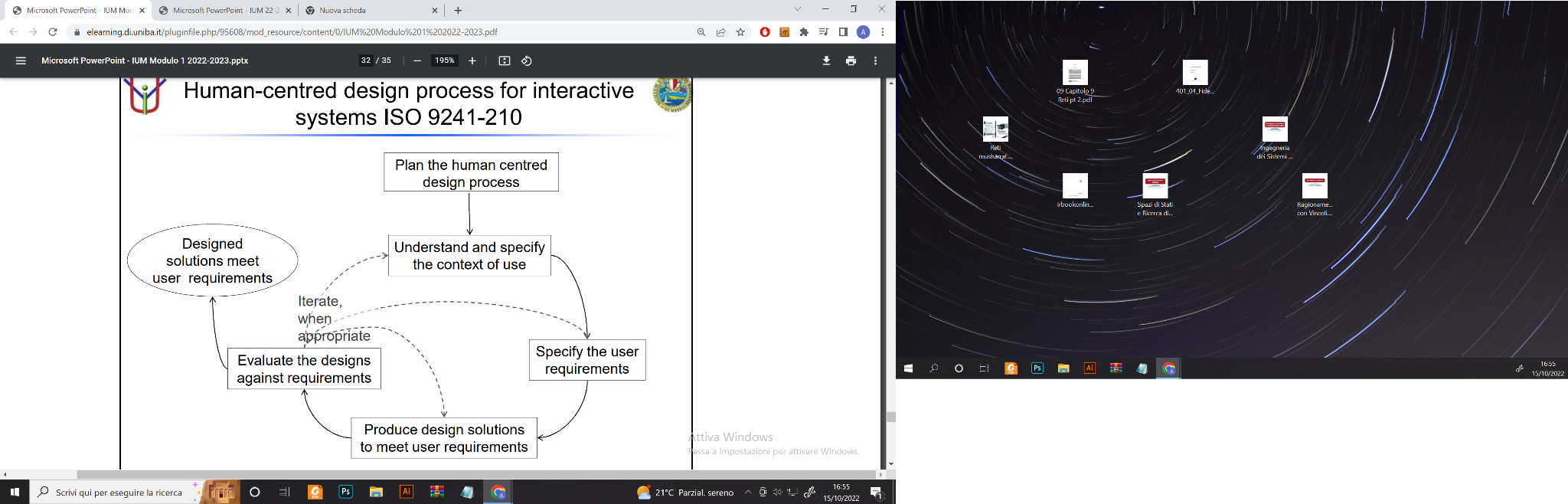
L’ISO 9241-210 (“*Human-Centered Design Process for interactive systems*”) (originariamente formava lo standard ISO 13407, ora aggiornato).

Per progettare un sistema usabile, il progettista deve porre l’attenzione sull’utente (studiarne le caratteristiche, le abitudini e le necessità in relazione all’uso del sistema), sui contesti in cui tale sistema sarà utilizzato e sui compiti svolti dall’utente.

Si parla di **progettazione centrata sull’essere umano** (*human-centred design*), un approccio generale caratterizzato da:

1. Il coinvolgimento attivo degli utenti e una chiara comprensione dei requisiti
2. un’assegnazione appropriata delle funzioni fra utenti e tecnologia
3. sviluppo iterativo in cui vengono creati e valutati (possibilmente con gli utenti) prototipi
4. progettazione multidisciplinare.

Nello standard ISO 9241-210, che ha lo scopo di “fornire una guida alle attività di progettazione centrata sull’essere umano lungo il ciclo di vita dei sistemi interattivi basati su computer”, viene fornito un modello iterativo per organizzare il processo di progettazione user-centered.



1. Fuori dal ciclo, la prima attività da svolgere è la pianificazione della progettazione human centred, che consiste nell’identificazione degli obiettivi del progetto e dei gruppi di utenti target.

* In particolare, identificare gli obiettivi di usabilità e UX fornendo pesi e priorità. A seconda del tipo di sistema da progettare serve identificare gli aspetti dell’usabilità (o della UX) da enfatizzare.

1. **Comprendere e specificare il contesto d’uso**: il contesto in cui il sistema sarà utilizzato è definito dalle caratteristiche degli utenti, dei compiti e dell’ambiente fisico e organizzativo. La descrizione del contesto d’uso dovrebbe comprendere

* Competenze, abilità esperienze, caratteristiche e abitudini degli utenti. Sarà spesso necessario classificare gli utenti in diverse categorie, in funzione dei diversi ruoli nei confronti del sistema. Il sistema deve avvicinarsi alle procedure utilizzate nelle pratiche quotidiane degli utenti.
* I compiti che gli utenti dovranno eseguire e il modo in cui questi possono influenzare l’usabilità del sistema.
* L’ambiente nel quale gli utenti utilizzeranno il sistema: caratteristiche dell’ambiente fisico e sociale, nonché le tecnologie utilizzate e le procedure di lavoro. Sulla base di ciò si decide su quali dispositivi implementare il sistema e si può porre attenzione alla criticità delle funzionalità.

1. **Specificare i requisiti utente e organizzativi**: si descrivono i requisiti in relazione al contesto d’uso individuato. Essi possono essere organizzati per livelli di priorità e formulati in modo da permettere la loro successiva convalida mediante opportuni test.
2. **Produrre soluzioni di progetto** che siano coerenti con i requisiti degli utenti: sviluppare proposte di progetto con un approccio multidisciplinare e realizzare prototipi e simulazioni. L’uso di prototipi permette ai progettisti di comunicare più efficacemente con gli utenti e riduce la necessità di costosi rifacimenti.

* Un prototipo serve a raccogliere le reazioni degli utenti, per poi utilizzarle nell’orientare le attività di progettazione successive.
* I prototipi sono inizialmente schizzi delle schermate del sistema.

1. **Valutare il progetto nei confronti dei requisiti:** il processo di valutazione dovrebbe essere pianificato, precisando quali parti del sistema devono essere valutate e come. La valutazione può essere usata per dimostrare che i requisiti human-centered sono stati raggiunti.

* Validazione sul campo: prova il funzionamento del sistema durante l’uso effettivo. A tale scopo si possono analizzare i dati raccolti dall’help desk e i feedback da utenti reali.
* Monitoraggio a lungo termine: alcuni effetti dell’utilizzo di un sistema non sono riconoscibili fino a che il sistema non sia stato utilizzato per un certo periodo di tempo.

La fase di valutazione in genere fa emergere problemi di varia natura: è necessario iterare il processo tornando a una delle fasi precedenti e, a seconda di quale sia la fase da riprendere, sarà necessario più o meno sforzo. Potrei dover riprogettare tutto il sistema così come dover aggiustare una specifica o semplicemente cercare una nuova soluzione per le specifiche non modificate.

L’utente è l’oggetto dell’interesse del progettista e, a seconda delle fasi in cui è coinvolto si distinguono diversi approcci:

* ***Progettazione centrata sull’utente****:* proposta da Norman e Draper, coinvolge l’utente nelle fasi di specifica dei requisiti e di valutazione.
* ***Participatory design*** o ***co-design***: coinvolge l’utente anche durante l’attività di progettazione.

**Confronto con progettazione system-centered**: lo human-centered design può essere considerato un approccio più maturo (non un’alternativa) rispetto al system-centred design. L’HCD contiene al suo interno le problematiche di questo ma le colloca in un contesto più ampio, che ci permette di comprendere in modo più approfondito le finalità del sistema.

### Sviluppo software a cascata

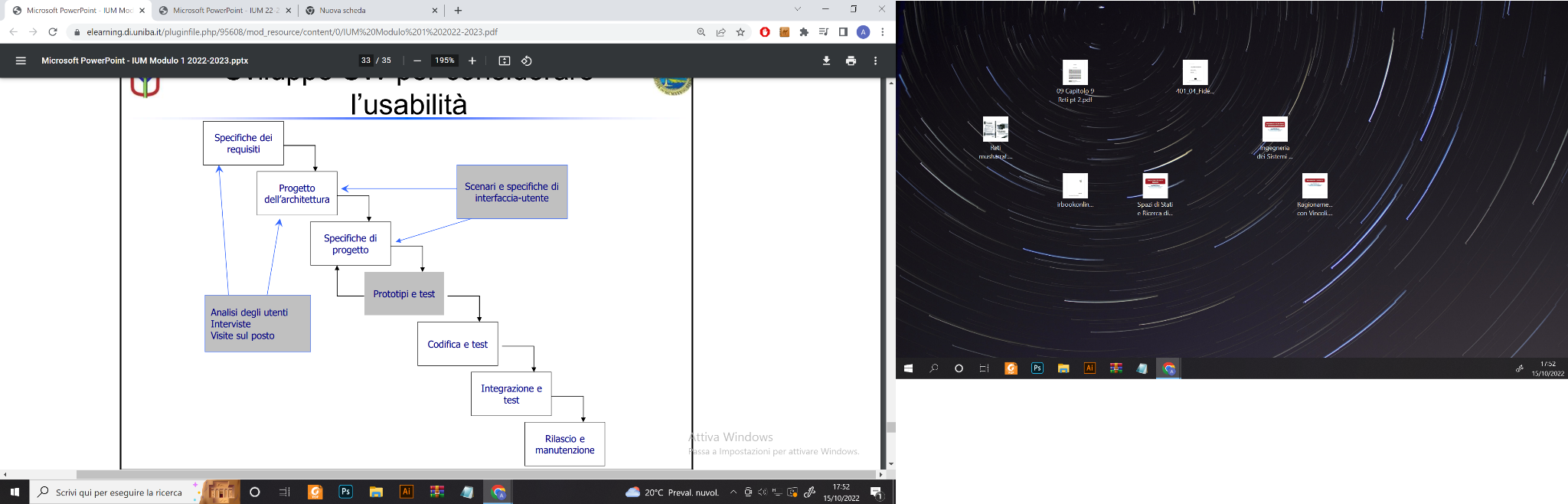
L’**ingegneria dell’usabilità** è la disciplina che studia le tecniche, i metodi e i processi che possono essere utilizzati per progettare e sviluppare sistemi usabili.

* Approccio pragmatico e basato su fondamenti scientifici
* Inizialmente focalizzata sul design della UI dei sistemi software, oggi comprende la totalità delle pratiche utilizzate nel processo di progettazione e sviluppo dei sistemi interattivi (a partire dalla raccolta e analisi dei requisiti).

I principi cardine della disciplina sono la focalizzazione sull’utente durante tutto il processo di progettazione, l’attuazione di prove con l’utente fornendo analisi qualitative e misure quantitative per la valutazione dell’usabilità e lo sviluppo iterativo (che avviene per prototipi successivi).

L’approccio visto supera il tradizionale modello di progettazione **a cascata**, che prevedeva lo svolgimento di tutte le fasi in modo puramente sequenziale. Tale modello ha fallito perché

* per nessun progetto reale le attività procedono in modo semplice e lineare: si rende spesso necessario tornare sui passi precedenti e rivedere decisioni considerate come consolidate.
* risulta impossibile specificare tutti gli aspetti di un sistema complesso all’inizio e poi realizzarlo senza modificare nulla
* difficoltà tecniche inattese possono non essere prevedibili e i requisiti possono essere stati non compresi appieno da chi li ha specificati.



Al contrario, nello sviluppo per l’usabilità serve che i prototipi e i test siano presenti già nelle prime fasi e il progetto deve anche includere l’interfaccia utente come aspetto principale e non secondario.

### Costi benefici

I costi dovuti alla messa in pratica di approcci human-centered sono d’investimento necessario per trasformare un’organizzazione di progetto tradizionale in un’organizzazione che utilizza i metodi dell’ingegneria dell’usabilità (attività di addestramento e di reclutamento di nuove risorse da inserire in team multidisciplinari). Potrebbe essere richiesta una prima fase di sperimentazione attraverso progetti pilota.

Difficilmente è attuabile in pratica condurre indagini per quantificare benefici economici attuali dovuti al cambio di paradigma; è possibile però condurre esperimenti concettuali effettuando opportune congetture per poi quantificare i guadagni ipotetici. Inoltre, ci sono importanti differenze di costi:

* Riduzione dei costi complessivi di progettazione e sviluppo derivanti da minori modifiche al prodotto rispetto che nelle fasi finali del progetto. I processi di progettazione iterativi permettono di anticipare i problemi di usabilità (assieme alle loro soluzioni), riducendo i costi di attuazione.
* Riduzione dei costi di manutenzione, per gli stessi motivi.
* Riduzione dei costi di supporto all’utente, in termini di formazione all’uso e documentazione tecnica più semplici.

Infine, si riscontra una maggiore soddisfazione dell’utente, con conseguente miglioramento dell’immagine del prodotto e della credibilità del fornitore sul mercato, con conseguente incremento dei volumi di vendita.